ALAUDDIN UNIVERSITY PRESS

Konsep *dan* Desain **Web**

Ridwan A. Kambau, S.T., M.Kom.

Ridwan A. Kambau, S.T., M.Kom.

Konsep & Desain WEB

Alauddin University Press

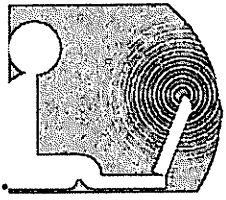
Konsep & Desain WEB

Penulis
Ridwan A. Kambau, S.T., M.Kom.

Penyunting:
Yusran Bobihu

x + 205, 14 x 21 cm
Cetakan I: Desember, 2012
ISBN: 978-602-237-370-4

Alamat:
Alauddin University Press
Jl. Sultan Alauddin No. 63 Makassar
Telp. 0823 4867 1117, Fax. 0411-864923
E-mail: au_press@yahoo.com



Sambutan Rektor

Tidakkah engkau malu pergi ke laut, sementara pulang hanya membawa sekendi air, padahal di dalam laut terdapat begitu banyak mutiara yang terpendam... demikian nasihat puitis Jalauddin Rumi dalam buku The Sufi Book of Life.

Syair inspiratif ini memberikan dorongan bagi siapa saja yang mengabdikan dirinya di dunia pendidikan apalagi di perguruan tinggi untuk menghasilkan dan melahirkan karya-karya akademik yang dapat memberikan pencerahan kepada siapapun. Sebuah ironi, jika orang-orang yang bergelut di dunia perguruan tinggi, ternyata hanya membawa sekendi "air" pengetahuan untuk mengobati dahaga masyarakat, padahal begitu banyak mutiara yang terpendam di dalamnya yang dapat memberi "sinar" kehidupan. Atas dasar inilah, ikhtiar untuk menjadikan kampus UIN Alauddin sebagai kampus peradaban harus terus digulirkan, sebab hanya kampus yang menjadikan orientasi "Peradaban" sebagai basis aktivititas dan tradisi keilmuannya yang akan mampu membawa semangat perubahan di tengah masyarakat menuju masyarakat madani.

Kampus peradaban yang dicita-citakan hanya bisa terwujud jika pengembangan kultur dan *mindset* akademik lebih relevan dengan suasana dan wadah yang bernama universitas Islam. Sebaliknya, jika orientasi peradaban hanya sebatas jargon dan simbol, maka status "universi-

tas" dan "Islam" akan menjadi beban bagi kita maupun masyarakat. Di satu sisi, UIN akan menjadi universitas pinggiran, sementara di sisi lain, karakter keislaman menjadi hilang. Karena itu, diperlukan usaha sungguh-sungguh untuk mengawal UIN Alauddin mencapai visi dan misinya untuk menjadi *world class university* yang berperadaban.

Untuk mencapai visi itu, maka program GSB (Gerakan Seribu Buku) ini menjadi salah satu langkah strategis memacu sivitas akademika untuk tidak sekadar meneguk "air" pengetahuan di perguruan tinggi, tetapi dapat membawa ribuan bahkan jutaan kendi "air dan mutiara" pengetahuan ke tengah masyarakat. Orang bijak berkata *"Buku adalah pengusung peradaban, tanpa buku sejarah menjadi sunyi, ilmu pengetahuan menjadi mati, dan kehidupan bisa kehilangan arti."*

Oleh karena itu, saya sangat bersyukur kapada Allah swt, atas terselenggaranya program GSB ini, baik tahun I maupun tahun II. Program GSB telah membuktikan kepada publik bahwa UIN Alauddin memiliki kekuatan dan potensi yang cukup besar untuk mewujudkan dan menghantarkan kampus ini menuju kampus peradaban melalui maha karya para civitas akademika. Melalui program GSB ini, potensi sumber daya UIN Alauddin akan terus digali, diapresiasi dan dihargai sehingga melahirkan kreasi, ide dan prestasi.

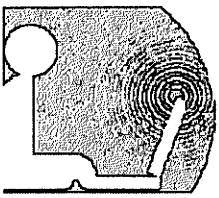
Selaku Rektor, saya senantiasa berharap agar *tagline* "Peradaban" yang selama ini digulirkan harus menjadi visi dan misi bersama yang tertanam dalam sebuah bingkai kesadaran kolektif bagi seluruh sivitas akademik untuk mewujudkan UIN Alauddin sebagai universitas yang kompetitif dan berkarakter. Untuk itu, tiga agenda besar; *pencerdasan, pencerahan* dan *prestasi* harus menjadi fokus perhatian utama bagi sivitas akademika UIN Alauddin. Ketiga agenda ini dirancang sebagai sebuah strategi untuk menjadikan UIN Alauddin lebih terbuka, dan menjadi pusat

kepeloporan pengembangan nilai dan akhlak serta keunggulan akademik-intelektual yang dipadukan dengan pengembangan teknologi untuk membangun sebuah masyarakat yang berperadaban.

Akhirnya, perkenankan saya mengucapkan terimakasih dan penghargaan yang setinggi-tingginya kepada seluruh sivitas akademika UIN Alauddin Makassar yang telah mencurahkan pikiran dan tenaganya dalam menghasilkan karya akademik ini. Semoga gagasan yang dituangkan di dalam buku ini mampu menjadi "air" penyejuk dan pengobat dahaga bagi masyarakat yang haus akan pencerahan, dan dapat menjadi "mutiara" yang memberikan cahaya bagi peradaban.{"}

Samata, 1 Nopember 2012
Rektor

Prof. Dr. H. A. Qadir Gassing. HT, MS



Kata Pengantar

Dengan rasa syukur penulis haturkan kehadirat Allah SWT, Tuhan yang Maha Pengasih dan Maha Penyayang. Atas Taufik dan Hidayah-Nya Sehingga buku ini dapat terselesaikan.

Salawat dan Salam kami tujukan pula kepada Nabi Muhammad SAW. Yang telah berjuang menyampaikan risalah Islam kepada seluruh umat manusia.

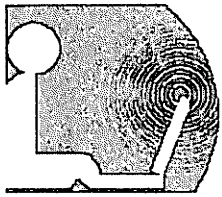
Ucapan terima kasih setinggi-tingginya penulis haturkan kepada Rektor Universitas Islam Negeri Alauddin Makassar, yang telah memberi kesempatan kepada penulis dalam mewujudkan buku ini melalui program gerakan seribu buku.

Adapun harapan penulis mudah-mudahan kehadiran buku ini menambah literatur yang berkaitan dengan Pendidikan. Penulis menyadari bahwa dalam buku ini masih terdapat kekurangan dan kekhilafan. Oleh karena itu kepada para pembaca dan para pakar, penulis mengharapkan saran dan kritik konstruktif demi kesempurnaan buku. Kepada penerbit dan semua pihak yang telah membantu penerbitan buku ini diucapkan terima kasih.

Wassalam

Makassar, 2012

Penulis

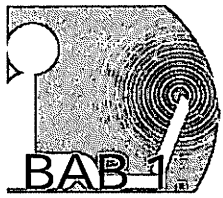


Daftar Isi

Web Klien ___ 41

Web Server ___ 67

Desain Web ___ 97



Gambaran Umum WEB

Menyediakan sebuah konsep pemahaman yang mendasar tentang teknologi web. Bagian ini mencakup:

- Pengertian Web
- Perkembangan Teknologi Web
- Komponen dan Semantik Web
- Uniform Resource Identifier (URI)
- HyperText Markup Language (HTML)
- HyperText Transfer Protocol (HTTP)

1.1 Pengertian Web

Web merupakan aplikasi yang bersifat request response layanan antara client dan server yang dijalankan pada protokol HTTP dalam suatu jaringan komunikasi data dengan menggunakan default port 80. Proses request response diawali dengan cara memasukkan URL yang sesuai dengan skema URI. Pada sisi klien, aplikasi yang digunakan dapat berupa browser untuk meminta dan menerima layanan, sedangkan pada sisi *server* dapat berupa *DNS Server, Web Server* dan *Database Server*. Port yang digunakan pada protokol ini bukanlah port yang berbentuk fisik, melainkan port yang bersifat logis.

Situs web(bahasa Inggris: *web site*) atau sering disingkat dengan istilah situs adalah sejumlah halaman web yang

memiliki topik saling terkait. Halaman tersebut memuat script (bahasa pemrograman) yang dapat memanggil file-file atau layanan dalam web server itu sendiri, dari server lain ataupun data yang berada dalam database. File-file atau layanan tersebut dapat berupa berkas-berkas gambar, video, atau jenis-jenis berkas lainnya. Data situs web biasanya ditempatkan pada sebuah web server dan database server yang dapat diakses melalui browser dalam suatu jaringan komputer seperti internet ataupun *Local Area Network*(LAN). Meskipun sebuah situs web dapat diakses publik secara bebas, pada prakteknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya, beberapa situs web mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota atau bahkan meminta pembayaran untuk dapat menjadi anggota. Pembatasan-pembatasan ini umumnya dilakukan karena alasan keamanan, menghormati privasi, atau karena tujuan komersil tertentu.

Halaman web merupakan berkas yang ditulis sebagai berkas teks biasa (plain text) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis HTML, XHTML ataupun XML. Halaman-halaman tersebut akan diakses oleh klien melalui aplikasi peramban web (browser). Berkas tersebut kemudian diterjemahkan oleh browser, selanjutnya ditampilkan pada layar (screen) komputer. Implementasi dan mekanisme pengaksesan dapat berupa skema http atau https. Pengaksesan melalui https digunakan untuk meningkatkan aspek keamanan dan aspek privasi yang lebih baik.

Website sebagai kumpulan dari halaman dapat bersifat statis maupun dinamis, yang membentuk satu rangkaian yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (hyperlink). Bersifat statis apabila isi informasi website tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik website. Bersifat

dinamis apabila isi informasi website selalu berubah-ubah dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna website.

1.2 Perkembangan Teknologi Web

Pada saat penulisan buku ini, dapat diketahui bahwa perkembangan teknologi web telah mencapai web 3.0. Salah satu cara termudah mengidentifikasi perkembangan teknologi web adalah dengan melihat konten halaman yang ditampilkan pada browser. Perbandingan teknologi web mulai dari awal pengembangannya sampai web 3.0 dapat dipaparkan sebagai berikut.

Web 1.0

Web 1.0 merupakan teknologi web yang pertama kali digunakan dalam aplikasi world wide web atau ada yang menyebut web 1.0. sebagai www itu sendiri yang banyak digunakan dalam situs web yang bersifat personal.

Beberapa ciri atau karakteristik web 1.0. adalah:

1. Halaman web besifat statis
2. Interaksi hanya bersifat satu arah antara pengunjung dengan web site
3. Form-form dikirim melalui e-mail.

Web 2.0

Web 2.0, adalah sebuah istilah yang dicetuskan pertama kali oleh O'Reilly Media pada tahun 2003 dan dipopulerkan pada konferensi web 2.0 pertama di tahun 2004, merujuk pada generasi kedua layanan berbasis web seperti situs jaringan sosial, wiki, perangkat komunikasi, dan folksonomi. Aplikasi web pada tahap ini menekankan pada kolaborasi online dan berbagi antar pengguna. O'Reilly Media dengan kolaborasinya bersama MediaLive International menggunakan istilah ini sebagai judul untuk sejumlah seri

konferensi. Istilah ini menunjukkan bahwa adanya versi baru untuk web. Penggunaan istilah ini hanya mengacu kepada cara pengembangan sistem di dalam menggunakan platform web, jadi bukan mengacu pada pembaruan spesifikasi teknis World Wide Web.

Tim Oreilly mendefinisikan istilah Web 2.0, sebagai berikut:

> "Web 2.0 adalah sebuah revolusi bisnis di dalam industri komputer yang terjadi akibat pergerakan ke internet sebagai platform dan suatu usaha untuk mengerti aturan-aturan agar sukses di platform tersebut."

Dari pendefinisian tersebut didapatkan beberapa prinsip-prinsip Web 2.0, yakni:

1. **Web sebagai platform** ***(The Web as Platform)***

   Aplikasi Web 2.0 menggunakan Web (atau Internet) sebagai platformnya. Yang dimaksud dengan platform pada penulisan ini adalah tempat suatu aplikasi dijalankan. Contoh platform yang terkenal adalah Windows, di mana ada aplikasi-aplikasi seperti Microsoft Office dan Adobe Photoshop. Menggunakan Internet sebagai platform berarti aplikasi-aplikasi tersebut dijalankan langsung di atas Internet dan bukan di atas satu sistem operasi tertentu. Contohnya adalah Google yang bisa diakses dari sistem operasi mana pun. Contoh lainnya adalah Flickr yang juga bisa diakses dari sistem operasi mana pun.

2. **Data sebagai pengendali utama** ***(Data is the Next Intel Inside)***

   Kekuatan aplikasi Web 2.0 terletak pada data. Aplikasi-aplikasi Internet yang berhasil selalu didukung oleh basis data yang kuat dan unik. Contohnya adalah Google, yang kekuatannya terletak pada pengumpulan dan manajemen data halaman-halaman Web di internet.

3. **Harnessing Collective Intelligence**

   Aplikasi Web 2.0 memiliki sifat yang unik, yaitu memanfaatkan kepandaian dari banyak orang secara kolektif. Sebagai hasilnya muncullah basis pengetahuan yang sangat besar hasil gabungan dari pengetahuan banyak orang. Contoh yang jelas adalah Wikipedia. Wikipedia adalah ensiklopedi online yang memperbolehkan semua orang untuk membuat dan mengedit artikel. Hasilnya adalah ensiklopedi online besar yang sangat lengkap artikelnya, bahkan lebih lengkap daripada ensiklopedi komersial seperti Encarta.

4. **End of the Software Release Cycle**

   Aplikasi Web 2.0 memiliki sifat yang berbeda dengan aplikasi pada platform seperti Windows. Suatu aplikasi Windows biasanya dirilis setiap dua atau tiga tahun sekali, misalnya saja Microsoft Office yang memiliki versi 97, 2000, XP, dan 2003. Di lain pihak, aplikasi Web 2.0 selalu terupdate secara terus menerus hal disebabkan oleh sifatnya yang bukan lagi produk melainkan layanan. Google misalnya, data dan programnya selalu terupdate tanpa perlu menunggu waktu-waktu tertentu.

5. **Lightweight Programming Models**

   Aplikasi Web 2.0 menggunakan teknik-teknik pemrograman yang mudah dan ringan untuk mendukung layanan berbasis service oriented application seperti AJAX dan RSS. Ini memudahkan orang lain untuk memakai ulang layanan suatu aplikasi Web 2.0 guna membentuk layanan baru. Contohnya adalah Google Maps yang dengan mudah dapat digunakan orang lain untuk membentuk layanan baru. Sebagai hasilnya muncullah layanan-layanan seperti HousingMaps yang menggabungkan layanan Google Maps dengan Craigslist. Layanan seperti ini, yang menggabungkan layanan dari aplikasi-aplikasi lainnya, dikenal dengan istilah mashup.

6. **Software Above the Level of a Single Device**

   Aplikasi Web 2.0 bisa berjalan secara terintegrasi melalui berbagai device. Contohnya adalah iTunes dari Apple yang berjalan secara terintegrasi mulai dari server Internet (dalam bentuk toko musik online), ke komputer pengguna (dalam bentuk program iTunes), sampai ke mobile device (dalam bentuk iPod).

7. **Rich User Experiences**

   Aplikasi Web 2.0 memiliki user interface yang kaya meskipun berjalan di dalam browser. Teknologi seperti AJAX memungkinkan aplikasi internet memiliki waktu respons yang cepat dan user interface yang intuitif mirip seperti aplikasi Windows yang di-install di pada koputer yang berbasis *dekstop*. Contohnya adalah Gmail, aplikasi email dari Google yang memiliki user interface revolusioner. Contoh lainnya lagi adalah Google Maps yang meskipun berjalan dalam browser namun bisa memberikan respons yang cepat saat pengguna menjelajahi peta.

Web 3.0

Teknologi Web generasi ketiga yang pertama kali diperkenalkan tahun 2001 ini memiliki ciri-ciri umum seperti suggest, happen dan provide dimana web seolah-olah sudah seperti kehidupan di alam nyata. Web 3.0 sendiri juga merupakan sebuah realisasi dari pengembangan sistem kecerdasan buatan (artificial intelegence) untuk menciptakan global metadata yang dapat dimengerti oleh sistem, sehingga sistem dapat mengartikan kembali data tersebut kepada pengunjung dengan baik.

Saat ini adaptasi Web 3.0 mulai dikembangkan oleh beberapa perusahaan di dunia seperti secondlife, Google Co-Ops, bahkan di Indonesia sendiri juga sudah ada yang mulai mengembangkannya, yaitu Li'L Online (LILO) Community.

Pada era web 3.0 pengembangan aspek sosial sebuah web mulai dipertimbangkan. Aspek sosial yang dimaksud adalah aspek interaksi, bagaimana sebuah web dapat memberikan sebuah interaksi sesuai dengan kebutuhan informasi setiap pemakaianya.

Sebagai teknologi masa depan, Web 3.0 juga membutuhkan kecepatan akses Internet yang memadai dan spesifikasi komputer yang lebih handal, hal ini disebabkan karena teknologi ini secara visual berbasis 3D. Web era ini bisa dibilang sangat *care* dengan kebutuhan kita karena menyediakan apa saja yang kita butuhkan. Contoh sederhana dengan dukungan teknologi 3-D animasi, kita bisa membuat profil avatar sesuai karakter kita kemudian melakukan aktivitas di dunia maya layaknya kehidupan sehari-hari kita di dunia nyata, mulai dari jalan-jalan ke mall, ke book store, bercakap-cakap dengan teman, dan sebagainya. Web 3.0 akan benar-benar menciptakan sebuah dunia virtual bagi manusia. Dia mampu memberi saran dan nasehat sekaligus menyediakan data dan informasi sesuai dengan kebutuhan pengguna. Memang, ini menjadi salah satu keunikan dari Web 3.0 karena konsep dasar yang digunakannya adalah manusia dapat berkomunikasi dengan mesin pencari. Misalnya kita bisa meminta Web mencari suatu data spesifik tanpa perlu kita susah payah mencari satu per satu dalam situs-situs Web. Hasil yang diberikan pun juga relevan.

Konsep Web 3.0 diarahkan sebagai web masa depan, tetapi masa depan yang bagaimana yang ditawarkan? Hal inilah yang terkadang masih menjadi perdebatan, tetapi bisa jadi tidak ada yang salah, karena mungkin sebagian besar pemikiran mengenai web masa depan tersebut sangat mungkin untuk terlaksana.

Ciri utama dari web 3.0 antara lain:

- Realisasi Semantik Web.

  Semantik web cukup dipercaya sebagai wujud dari Web 3.0, dengan kecerdasan buatan, Web 3.0 diharapkan akan merealisasikan konsep Semantik web dan menjadi generasi selanjutnya dari WWW.

- Evolusi 3D.

  Tidak mengherankan bahwa kemampuan 3D selalu merupakan cerminan masa depan, evolusi 3D telah terjadi pada game animasi, dan lain-lain, walaupun saat ini masih belum mengubah mayoritas wajah web. Tampilan 3D bisa jadi memang dihindari oleh sebagian pengakses Internet karena tampilan dan proses 3D berarti pula pertukaran data yang lebih besar dan tentu berpengaruh pada kecepatan maupun biaya yang dikeluarkan. Tentunya, evolusi 3D ini hanya akan berhasil jika infrastruktur di masa mendatang telah mendukung pengguna Internet pada umumnya.

- Web sebagai Database.

  Masih sering kita dengar istilah web statik dan web dinamis, Skema OWL. Web statik menunjukkan bahwa website tersebut selalu memberikan informasi yang sama sebagai respon pada setiap pengunjung yang mengaksesnya. Sementara web dinamis merupakan kebalikannya, di mana informasi yang diberikan website tersebut dapat berubah secara interaktif tergantung pada kondisi dan konteks yang distimulasikan oleh pengguna. Pada Web 3.0, diharapkan website merupakan database dan tentunya semakin interaktif dan dinamis kepada pengunjung, atau dinamakan dengan Data Web. Salah satu teknologi yang dikembangkan adalah SPARQL yang menyediakan bahasa query standard dan Application Programming Interface (API) untuk menelusuri database RDF yang terdistribusi pada website.

- Executable.

  Jika kita melihat kembali perjalanan web dimulai dari Web 1.0, maka dapat dikatakan bahwa pengunjung Web 1.0 hanya memiliki hak sebatas read-only, karena sebagai pengunjung user hanya akan membaca informasi yang ditampilkan Web 1.0. Tidak heran jika kemudian istilah yang sering dipakai saat mengakses Internet adalah "browsing", fungsi browser Internet sebatas untuk melihat informasi dari satu website ke website lainnya. Pada Web 2.0, sebagai pengunjung Anda dapat melakukan kontribusi dan memiliki hak untuk readwrite, di mana Anda dapat berperan aktif pada website tersebut. Istilah "sharing" mulai umum digunakan dalam konsep Web 2.0. Web 3.0, menambah lagi hak Anda menjadi executable mengizinkan Anda memodifikasi website itu sendiri. Dapat disimpulkan untuk mewujudkan Web 3.0, maka harus didukung oleh kemampuan dan teknologi yang merealisasikan transformasi dari web yang terpisah secara aplikasi dan penyimpanan data, menjadi saling berinteraksi sesama mesin. Interaksi tidak hanya terjadi antara pengunjung dan website, tetapi juga di antara website itu sendiri dalam formatnya sendiri. Istilah World Wide Web bisa jadi berubah menjadi World Wide Database untuk menunjukkan database yang terdistribusi dan dimungkinkan dengan adanya teknologi yang mendukung Semantik web.

1.3 Komponen Semantik Web

Istilah semantik web sendiri telah lebih dulu dikenal dibandingkan istilah Web 3.0. Semantik web ditampilkan tidak hanya dalam format bahasa manusia yang umum (natural language), tetapi juga dalam format yang dapat dibaca dan digunakan oleh mesin (software). Seperti yang kita ketahui, website ditujukan untuk memberikan informasi

kepada manusia. Misalnya saat menginginkan sebuah buku, pengguna dapat menelusurinya pada *search engine* atau website tertentu hingga akhirnya mendapatkan buku tersebut. Misalkan terdapat pilihan dari berbagai kategori untuk mendapatkan buku yang dimaksud, mesin sendiri tidak dapat memutuskan dan melakukannya tanpa arahan dari manusia karena informasi tersebut diperuntukkan agar dimengerti hanya oleh manusia dengan menggunakan natural language. Kondisi inilah yang ingin diubah oleh Semantik web. Semantik web akan memiliki informasi yang dimengerti oleh mesin yang memiliki kecerdasan buatan hingga mampu menemukan dan mengintegrasikan informasi dengan mudah. Dengan demikian fungsi web menjadi wadah universal bagi pertukaran data, informasi, dan pengetahuan yang dapat menghasilkan kecerdasan buatan dan dapat mengerti keinginan pengguna. Semantik web dapat diinstruksikan untuk mengambil informasi sesuai kriteria tertentu. Beberapa format dan spesifikasi yang dikenal oleh mesin dalam Semantik web antara lain adalah RDF (Resource Description Framework) dan OWL (Web Ontology Language).

RDF

Salah satu "tulang punggung" pada teknologi Web 3.0 adalah format dan spesifikasi yang memungkinkan komunikasi dan interaksi pada level mesin. W3C (*World Wide Web Consortium*) mendefinisikan format metadata yang dikenal dengan RDF (*Resource Description Framework*).

RDF terdiri dari tiga komposisi yaitu:

- Subject
- Predicate
- Object

Predicate merupakan komposisi yang menerangkan sudut pandang dari subject yang dijelaskan object, sementara

subject dan object merupakan entitas. Object di dalam RDF dapat menjadi subject yang diterangkan oleh object yang lainnya. Dengan inilah object dapat berupa masukan yang dapat diterangkan secara jelas dan detail, sesuai dengan keinginan pengguna yang memberikan masukan. Cara kerja RDF dapat diterangkan dengan satu contoh sederhana berikut, untuk mendefinisikan "daun memiliki warna hijau", maka "daun" direpresentasikan sebagai subject, "hijau" merupakan object, dan "memiliki warna" adalah predicate. Dengan menggunakan RDF, website dapat menyimpan dan melakukan pertukaran informasi antar-web.

Aplikasi-aplikasi yang menggunakan RDF, antara lain:

- RSS (*RDF Site Summary*).

  RSS memberikan informasi update sebuah website tanpa pengunjung perlu mengunjungi website tersebut.
- FOAF (*Friend of a Friend*).

  Didesain untuk mendeskripsikan orang-orang, ketertarikan dan hubungan mereka.
- SIOC (*Semantikally-Interlinked Online Communities*).

  Menerangkan komunitas online dan menciptakan koneksi antara diskusi berbasis Internet seperti message board, blog maupun mailing list.

OWL

Ontology Web Language atau OWL didesain agar dapat digunakan oleh aplikasi yang memroses informasi OWL berbasis XML dan dapat dengan mudah dipertukarkan antara mesin dengan sistem operasi yang berbeda dan bahasa aplikasi yang berbeda.

OWL memiliki tiga sub-language (spesies), yaitu:

- OWL Lite.

Mendukung pengguna yang memerlukan klasifikasi hirarki dan dalam batasan yang sederhana.

- OWL DL.

  Mendukung konstruksi seluruh OWL, tetapi hanya dapat digunakan pada batasan tertentu.

- OWL Full.

  Diperuntukkan bagi pengguna yang menginginkan maksimum penggunaan dan kebebasan sintaksis.

1.4 URI (*Uniform Resource Identifier*)

1.4.1 Pengenalan URI

Sebuah URI menyediakan sarana sederhana dan extensible untuk mengidentifikasi sumber daya pada aplikasi web. Spesifikasi sintaks dan semantik URI berasal dari konsep yang diperkenalkan oleh World Wide Web, yang dijelaskan dalam "Identifier Universal Resource di WWW "[RFC1630]. Sintaks ini dirancang untuk memenuhi rekomendasi yang tercantum dalam *Rekomendasi Fungsional untuk Internet Resource Locators* dan *Fungsional Persyaratan Untuk Penyeragaman Nama Sumber Daya.*

Penjabaran dari URI adalah sebagai berikut:

Seragam (Uniform)

Keseragaman memberikan beberapa manfaat. Hal ini memungkinkan berbagai jenis sumber daya akan pada konteks yang sama, walaupun ketika mekanisme yang digunakan untuk mengakses sumber daya tersebut berbeda.

Sumber (Resource)

Spesifikasi ini merujuk pada sumber daya digunakan dalam pengertian umum untuk apa pun yang mungkin diidentifikasi dengan URI. Contoh umum mencakup

dokumen elektronik, gambar, sumber informasi dengan tujuan yang konsisten (misalnya, "Laporan Cuaca untuk Makassar hari ini "), sebuah layanan (misalnya, Gateway HTTP-to SMS), dan koleksi sumber daya lainnya. Sebuah sumber daya tidak selalu diakses melalui Internet, misalnya, manusia, perusahaan, dan buku yang terikat di perpustakaan juga dapat menjadi sumber daya. Demikian juga, konsep-konsep abstrak dapat menjadi sumber daya, seperti operator dan operan dari persamaan matematika atau nilai numerik (misalnya, nol, satu, dan tak terhingga).

Pengidentifikasian (Identifier)

Identifier mewujudkan informasi yang diperlukan untuk membedakan apa yang diidentifikasi dalam ruang lingkup identifikasi. Tujuan membedakan satu sumber daya dengan sumber daya lainnya, tidak terlepas dari bagaimana tujuan pembentukan sumber daya tersebut. Istilah-istilah ini tidak boleh keliru sebagai sebuah asumsi bahwa sebuah identifier mendefinisikan atau mewujudkan identitas apa yang direferensikan, meskipun itu mungkin terjadi untuk beberapa pengidentifikasian. Dalam banyak kasus, URI digunakan untuk menunjukkan sumber daya tanpa melihat konten apa yang akan mereka sediakan. Demikian juga, sumber daya diidentifikasi mungkin tidak tunggal di alam (misalnya, sumber daya mungkin bernama set atau pemetaan yang bervariasi dari waktu ke waktu).

URI adalah sebuah identifier yang terdiri dari urutan karakter dengan melakukan pencocokan aturan sintaks yang dapat memungkinkan terciptanya keseragamaman identifikasi sumber daya. URI memiliki lingkup global dan diinterpretasikan secara konsisten terlepas dari interpretasi dalam kaitannya dengan konteks pengguna akhir. Misalnya,

"http://localhost/" memiliki interpretasi yang sama sebagai acuan pada setiap pengguna bahkan meskipun interface jaringan sesuai dengan "localhost" mungkin berbeda untuk setiap pengguna akhir. URI yang mengidentifikasi dalam kaitannya dengan konteks lokal pengguna akhir hanya digunakan ketika konteks sendiri adalah aspek yang menentukan sumber daya, seperti ketika sebuah bantuan manual on-line merujuk ke file di sistem file pengguna akhir(misalnya, "file :/// contoh/hosts").

1.4.2 Sintaks URI

Setiap URI dimulai dengan nama skema yang mengacu pada sebuah spesifikasi untuk pengidentifikasian dan penetapan skema. Penulisan skema nama URI seperti "http", "ftp", "mailto" atau "file" diikuti dengan titik dua karakter dan kemudian oleh sebuah bagian skema tertentu. Dengan demikian, sintaks URI adalah penamaan federasi dan extensible sistem dimana spesifikasi masing-masing skema lebih lanjut dapat membatasi sintaks dan semantik dari skema tersebut.

Contoh URI berikut menggambarkan skema URI beberapa variasi dalam komponen umum sintaks:

- mailto:andry@contoh.org
- ftp://contoh.org/namadirektori/namafile
- news:comp.infosystems.www
- tel:+1-816-555-1212
- ldap://ldap.contoh.org/c=GB?objectClass?satu
- urn:oasis:names:tc:entity:xmlns:xml:catalog

1.4.3 Pertimbangan Desain URI

Sintaks URI dirancang dengan transkripsi global sebagai salah satu pertimbangan utamanya. URI adalah urutan karakter yang sangat terbatas yang dapat memuat: huruf dari abjad Latin dasar, angka, dan karakter khusus. URI

dapat digambarkan dalam dunia nyata seperti tinta di atas kertas, piksel pada layar, atau urutan oktet karakter encoding. Penafsiran URI hanya bergantung pada karakter yang digunakan dan bukan pada bagaimana karakter direpresentasikan dalam sebuah protokol jaringan.

Tujuan dari transkripsi dapat digambarkan oleh sebuah skenario sederhana. Bayangkan dua rekan kerja Amir dan Abdillah, duduk di sebuah meja pada sebuah konferensi internasional dan bertukar ide penelitian. Amir meminta Abdillah mengenai lokasi untuk mendapatkan informasi lebih lanjut, jadi Abdillah menulis URI untuk penelitian situs pada secarik kertas. Setelah kembali di rumah, Amir mengambil kertas tersebut dan memasukkan URI ke komputer, yang kemudian mengambil informasi yang disebutkan oleh Abdillah.

Beberapa pertimbangan dalam skenario desain sintaks URI adalah sebagai berikut:

- URI adalah urutan karakter yang tidak selalu mewakili urutan oktet.
- Sebuah URI dapat ditranskripsi dari sumber non-jaringan, dengan demikian harus terdiri dari karakter yang paling mungkin dapat dimasukkan ke dalam komputer dalam batasan yang dikenakan oleh keyboard (dan perangkat input yang terkait) di seluruh bahasa dan perangkat.
- URI harus lebih mudah diingat oleh manusia

1.4.4 Notasi Sintaks

Spesifikasi penulisan sintaks menggunakan notasi Augmented Backus Naur Form-(ABNF). Inti dari sintaks ABNF adalah didefinisikan dengan ALPHA (huruf), CR (carriage return), Delapan Angka (angka desimal), DQUOTE (kutipan ganda), HEXDIG (digit heksadesimal), LF (line feed), dan SP (spasi).

1.4.5 Komponen Sintaks

Sintaks URI terdiri dari urutan hirarkis komponen disebut sebagai skema, otoritas, path, query, dan fragmen.

URI = skema ":" "?" hier-part [query] ["#" fragmen]

hier-part = "/ /" otoritas path

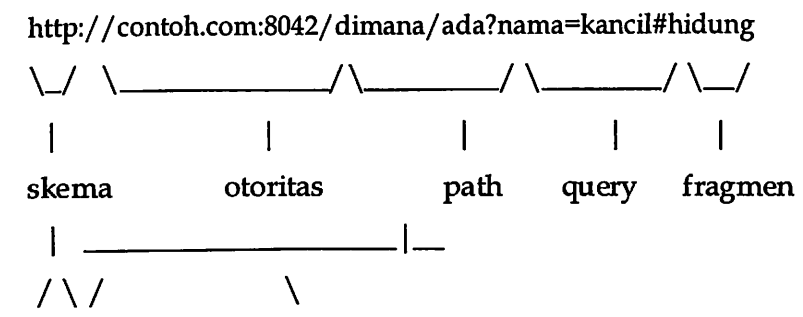
/ Path absolut-

/ Path-tanpa akar

/ Path-kosong

Skema dan komponen path sangat diperlukan, meskipun pathnya mungkin kosong (tidak ada karakter). Ketika adanya otoritas, maka path harus kosong atau dimulai dengan karakter garis miring ("/"). Sementara ketika otoritas tidak ada, maka path tidak dapat diawali dengan karakter dua garis miring ("/ /"). Hal ini mengakibatkan pembatasan dalam lima urutan ABNF yang tentunya tidak sesuai dengan aturan penulisan path.

Berikut ini adalah dua contoh URI dan komponen-komponennya:



Skema

Setiap URI dimulai dengan nama skema yang mengacu pada sebuah spesifikasi untuk menetapkan pengidentifikasian

dalam skema tersebut. Nama Skema terdiri dari urutan karakter yang dimulai dengan huruf dan diikuti dengan kombinasi huruf, angka, ditambah ("+"), Titik ("."), Atau tanda hubung ("-"). Meskipun skema adalah case-tidak sensitif, akan tetapi bentuk kanonik dan penulisan dokumen yang menentukan skema sebaiknya ditulis dengan huruf kecil hal ini dianjurkan untuk menjaga konsistensi penamaan. Pada implementasinya huruf besar dipandang setara dengan huruf kecil dalam penamaan sebuah skema (misalnya, "HTTP" dianggap sama dengan "http").

skema ALPHA = * (ALPHA / digit / "+" / "-" "." /)

Otoritas

Kebanyakan skema URI memuat hirarki elemen untuk penamaan sebuah otoritas agar menemukan spase(sumber daya) sesuai dengan yang didefinisikan. Sintaks asli secara umum menyediakan sarana untuk membedakan otoritas berdasarkan daftar nama atau alamat server dan dengan pilihan port yang digunakan serta informasi pengguna. Komponen otoritas didahului oleh sebuah karakter garis miring ganda ("/ /") dan diakhiri oleh karakter garis miring berikutnya ("/"), tanda tanya ("?"), atau karakter tanda pagar ("#") pada akhir URI.

otoritas = [userinfo "@"] host [":" port]

Path

Komponen path berisi data biasanya diorganisir dalam bentuk hirarki bersama dengan data dalam komponen query yang non-hirarki. Sebagaimana pada contoh di atas bahwa path itu diakhiri oleh tanda tanya pertama ("?") atau karakter tanda pagar ("#"), atau pada akhir URI. Path terdiri dari urutan segmen yang dipisahkan oleh karakter garis miring ("/"). Sebuah path selalu ditetapkan untuk URI, meskipun

path yang didefinisikan mungkin kosong (nol panjang). Penggunaan karakter diperlukan ketika sebuah URI akan digunakan sebagai konteks untuk referensi relatif.

Misalnya:

URI <mailto:sesuatu@contoh.com> memiliki path "sesuatu@contoh.com",

sedangkan <http://info.contoh.com?sesuatu>memiliki path URI yang kosong.

Query

Komponen query diawali oleh penggunaan karakter tanda tanya ("?") dan diakhiri dengan karakter tanda pagar ("#") atau pada akhir URI.

query = * (pchar / "/" / "?")

Karakter slash ("/") dan tanda tanya ("?") dapat mewakili data dalam komponen query.

Fragmen

Identifikasi komponen fragmen URI memungkinkan identifikasi secara tidak langsung pada sumber daya sekunder dengan mengacu pada sumber daya primer dan informasi identitas tambahan. Komponen pengenal fragmen ditandai dengan adanya karakter tanda pagar ("#") dan diakhiri oleh akhir URI.

fragmen = * (pchar / "/" / "?")

Semantik dari sebuah identifikasi fragmen didefinisikan oleh representasi himpunan yang mungkin timbul dari tindakan pengambilan data pada sumber daya.

1.4.6 URI, URL dan URN

Sebuah URI dapat lebih diklasifikasikan sebagai pencari, nama, atau keduanya. Istilah *Uniform Resource Locator* (URL) mengacu pada bagian dari URI untuk meng-

identifikasi sumber daya, menyediakan sarana mencari sumber daya dengan menjelaskan mekanisme akses utama. Istilah *Uniform Resource Name* (URN) telah digunakan historis untuk merujuk kepada kedua skema URI, yang dibutuhkan untuk tetap unik secara global bahkan ketika sumber daya tidak ada lagi atau menjadi tidak tersedia.

1.5 Hyper Text Markup Language (HTML)

1.5.1 Pendahuluan

HyperText Markup Language (HTML) adalah format data sederhana yang digunakan untuk membuat dokumen hypertext yang portable dari satu platform ke platform yang lain. Dokumen HTML adalah dokumen SGML dengan semantik yang sesuai untuk mewakili informasi dari berbagai domain.

HTML telah digunakan oleh World Wide Web (WWW) sejak tahun 1990. Sebelumnya, dokumentasi informal pada HTML telah tersedia dan dimiliki oleh sejumlah sumber di internet. Kemudian spesifikasi HTML bersama-sama dibangun dan meluncurkan satu set fitur yang secara kasar sesuai dengan kemampuan HTML yang digunakan secara umum sebelum Juni 1994.

HTML adalah aplikasi dari Standar ISO 8879:1986:

"Information Processing Text and Office Systems; Standard Generalized Markup Language" (SGML).

HTML Document Type Definition (DTD) adalah definisi resmi dari sintaks HTML dalam hal SGML. Spesifikasi ini juga mendefinisikan HTML sebagai Internet Media Type [IMEDIA] and MIME Content Type [MIME], pendefinisian semantik dari sintaks HTML ini berguna pada cara penafsiran sintaks oleh browser.

1.5.2 HTML dan SGML

SGML adalah sebuah sistem untuk mendefinisikan type dokumen terstruktur dan bahasa markup. Istilah "HTML" mengacu pada definisi tipe dokumen dan bahasa markup.

Dokumen HTML adalah dokumen SGML yang memuat urutan karakter yang secara fisik menjadi satu set entitas dalam bentuk hirarki dari elemen logis. Dalam spesifikasi SGML, sintaks dari SGML sendiri terpisah menjadi tiga bagian: yaitu deklarasi, prolog, dan instance. Untuk spesifikasi tersebut, prolog adalah DTD. DTD ini menggambarkan penggunaan tata bahasa, dimana simbol awal diberikan dalam deklarasi DOCTYPE dan diakhiri dengan karakter data dan tag, hasil akhirnya bergantung pada deklarasi elemen itu sendiri. Dalam hal ini tentu harus sesuai dengan DTD, yaitu harus sesuai dengan tata bahasa yang telah didefinisikan dan disetujui bersama.

1.5.3 Struktur Dokumen

Item pertama yang muncul dalam kode sumber dari suatu halaman web adalah deklarasi DOCTYPE. Hal ini memberikan informasi kepada web browser (atau user agent lainnya) tentang informasi jenis bahasa markup dimana halaman tersebut ditulis. Deklarasi ini akan mempengaruhi cara browser merender konten tersebut. Pendeklarasian DOCTYPE mungkin akan "terkesan" menyulitkan ketika pembuatan sebuah halaman web, akan tetapi kabar baiknya bahwa sebagian besar editor WYSIWYG web akan menciptakan DOCTYPE secara otomatis setelah pemilihan jenis dokumen yang akan dibuat.

Contoh Pendeklarsian DOCTYPE:

<DOCTYPE html PUBLIC "- // W3C // DTD XHTML 1.0 Transitional // EN" "Http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd~~V">

Deklarasi di atas dapat di uraikan sebagai berikut:

- DOCTYPE dimulai dengan string <!DOCTYPE, yang harus ditulis dalam huruf besar.
- Bagian berikutnya berbunyi html, mengacu pada nama dari elemen root untuk dokumen. Informasi tersebut digunakan untuk tujuan validasi, karena DTD sendiri tidak mengatakan bahwa elemen adalah elemen root di pohon dokumen.
- Pernyataan PUBLIC memberitahu browser bahwa DTD adalah sumber daya yang tersedia untuk umum.
- Bagian selanjutnya dikenal sebagai Identifier Publik, dan menyediakan informasi tentang pemilik DTD dalam hal ini adalah W3C. Identifier umum, yang ditampilkan di sini *tidak bersifat case* sensitive, juga termasuk tingkat bahasa DTD yang mengacu pada XHTML (1,0), dan mengidentifikasi bahasa *DT-bukan* isi dari halaman web.
- EN adalah bahasa yang didefinisikan sebagai bahasa Inggris, akan tetapi referensi EN bukan berarti bahwa bahasa yang terdapat dalam halaman web adalah harus bahasa Inggris semuanya.
- Bagian akhir merupakan Formal Public Identifier (FPI). Akan tetapi, ketika DOCTYPE memuat sebuah URL, maka akan dikenal sebagai Formal System Identifier(FSI), yang mengacu pada lokasi dari DTD.

DOCTYPE akan terlihat seperti di bawah ini jika berada dalam sebuah halaman web (halaman sangat sederhana HTML tanpa isi situs).

```
<DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "Http://www.w3.org/TR/html4/strict.dtd~~V">
        <html>
        <head>
                <title> Judul halaman </ title>
```

```
</ head>
<body>
</ body>
</ html>
```

DOCTYPE pada contoh di atas, berhubungan dengan Strict HTML 4.01. dan tampak bahwa halaman tersebut merupakan sebuah pohon dokumen. Sebuah halaman web dapat dianggap sebagai pohon dokumen yang dapat berisi jumlah cabang dan terdapat aturan khusus untuk isi dari item pada masing-masing cabang. Pemahaman tentang konsep pohon dokumen, sangat berguna untuk mempertimbangkan sebuah halaman web yang akan dibangun.

1.5.4 Elemen Dokumen HTML

Elemen yang akan dibahas pada buku ini tentu elemen-elemen yang berada diantara tag <html> dengan tag </ html>. Elemen-elemen tersebut akan mempengaruhi isi dan tampilan sebuah halaman web pada jendela browser (viewport).

Contoh halaman HTML sederhana dan elemen-elemen yang terkandung di dalamnya:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//
EN">
<html>
<head>
<title>Konsep dan Desain Web</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style>
.gaya{
    font-family: "Times New Roman", Times, serif;
    font-size: 18px;
    color: #FFFFFF;
    background-color: #000000;
```

```
}
</style>
</head>
<body>
<p class="gaya"> Teks ini akan ditampilkan, sesuai dengan class gaya</p>
</body>
</html>
```

Contoh di atas jelas memberikan pemahaman bahwa elemen html pada kenyataannya mengandung dua elemen yaitu head dan body. Head memiliki dua cabang yaitu elemen meta dan title. Sedangkan elemen body berisi elemen paragraf(<p>). Perhatikan bahwa ada beberapa simetri dengan cara tag dibuka dan ditutup untuk setiap elemen.

1.5.4.1 html

Setelah deklarasi DOCTYPE maka elemen html ini akan menjadi elemen akar pada pohon dokumen dan segala sesuatu yang berikut adalah keturunan dari elemen tersebut (root). Jika elemen root ada dalam konteks dokumen yang sudah diidentifikasi oleh DOCTYPE sebagai XHTML, maka html elemen juga membutuhkan atribut (ini tidak diperlukan untuk dokumen HTML) xmlns (ruang untuk penamaan XML):

<html *xmlns="http://www.w3.org/1999/xhtml"*>

Berikut ini adalah contoh dari halaman transisi dari XHTML:

```
<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"Http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd~~V">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <title> Judul halaman </title>
 </head>
 <body>
 </ body>
</ html>
```

1.5.4.2 Head

Elemen head berisi meta data-informasi yang menjelaskan dokumen itu sendiri atau sumber daya terkait, seperti script dan style sheet.

Elemen head dapat berisi:

- title

  Title akan menjadi judul dokumen atau nama dasarnya atau yang mengidentifikasi isi dokumen. Konten di dalam title dapat digunakan untuk memberikan judul yang muncul di bar judul browser, dan ketika pengguna menyimpan halaman sebagai favorit. Ini juga merupakan bagian yang sangat penting dari informasi dalam hal memberikan ringkasan yang berarti halaman untuk search engine, yang menampilkan title konten dalam hasil pencarian.

- base

  mendefinisikan URL dasar untuk link atau sumber daya pada halaman, dan jendela sasaran untuk membuka konten terkait.

- link

  link akan mengacu pada sumber daya dari beberapa jenis, yang paling sering untuk sebuah style sheet yang menyediakan instruksi tentang bagaimana tampilan berbagai elemen pada halaman web

- meta

  meta memberikan informasi tambahan tentang halaman, misalnya penggunaan karakter encoding halaman menggunakan ringkasan dari konten halaman, instruksi ke mesin pencari tentang apakah konten akan diindeks atau tidak dan sebagainya

- object

  merupakan wadah untuk objek media.

- script

  digunakan baik untuk menanamkan atau baca skrip eksternal

- style

  menyediakan suatu tempat untuk mendefinisikan gaya (khusus halaman tersebut) dengan gaya CSS.

Semua elemen di atas bersifat opsional dan dapat muncul dalam urutan apapun dalam head. Tetapi pemilihan dan penggunaan dari elemen di atas sangat mempengaruhi konten pada halaman (semua yang didefinisikan di dalam elemen body).

1.5.4.3 body

Body merupakan tempat dari sebagian besar kandungan sebuah halaman web. Segala sesuatu yang terlihat di jendela browser (atau viewport) sebagian besarnya adalah yang terkandung di dalam elemen ini termasuk paragraf, link, gambar, tabel, dan lain-lain. Elemen body memiliki beberapa atribut yang unik.

Elemen-elemen yang terdapat dalam elemen body dapat berupa:

1. **Format Teks**

   Elemen Heading <H1> sampai <H6>

   Contoh penggunaan:

   <H1> Ini adalah heading H1</ H1>

   <H2> Ini adalah heading H1</ H2>

H1: Ukuran huruf sangat besar, tebal dan terpusat. Satu atau dua baris kosong di atas dan di bawah.

H2: Ukuran huruf besar dan tebal. Satu atau dua baris kosong atas dan di bawah.

H3: Ukuran Huruf besar, miring serta sedikit menjorok dari kiri margin. Satu atau dua baris kosong di atas dan di bawah.

H4: Ukuran huruf tebal, normal, indentasi lebih dari H3. Satu baris kosong di atas dan di bawah.

H5: Ukuran Huruf miring, normal, dan indentasi sebagai H4. Satu baris kosong di atas.

H6: Ukuran huruf tebal, indentasi yang sama dengan teks normal, lebih dari H5 dan satu kosong baris di atas.

2. Elemen Form

Form <form>

Elemen <form> berisi urutan elemen masukan, bersama dengan unsur-unsur penataan dokumen.

Atribut pada elemen form adalah:

- Id, name:
  menunjukan identitas dari form. Atribut id bisa bersifat optional.
- action:
  menentukan tindakan URI untuk formulir.
- metode:
  memilih metode pengiriman data melalui URI, bentuk metode dapat berupa POST dan GET.
- enctype:
  menentukan jenis media yang digunakan untuk meng-kodekan nama/nilai yang akan dikirimkan melalui protokol.

Contoh penggunaan elemen form:

<form id="gaya" name="namaform" action="halamanref.html" method="get"

enctype="multipart/form-data"> </form>

Field Input <input type>

Elemen <input> merupakan field yang dapat digunakan untuk memasukan data pada halaman web dan berada di dalam elemen form. Ada beberapa penggunaan atribut yang

membedakan antara beberapa jenis input, perbedaan ini biasanya ditimbulkan oleh nilai dan type atributnya.

a. Input type = text

   Nilai default dari atribut adalah text, menunjukkan penyediaan inputan untuk satu baris teks. Sementara jika ingin memasukan teks yang memuat multibaris, maka elemen yang digunakan adalah <textarea>.

   Atribut pada elemen input, adalah:

- name:
  nama untuk elemen input
- MaxLength:
  membatasi jumlah karakter yang dapat dimasukkan ke dalam kolom teks masukan. (atribut Maxlength bersifat optional).
- Size:
  menentukan jumlah ruang pada layar yang dialokasikan untuk bidang inputan. Default untuk tampilan tergantung pada konfigurasi browser. (atribut Maxlength bersifat optional).
- Value:
  nilai(data) awal pada form inputan.

  Contoh penggunaannya:

  Username : <input type="text" name="namaInput" size="25" maxlength="25">

b. Input type = password

   Elemen <input type=password> pada dasarnya memiliki atribut yang hampir sama dengan type text, akan tetapi kolom teks pada field inputan ditampilkan dengan sebuah nilai yang "dikaburkan". Untuk menjaga privasi dan keamanan data inputan dengan type password, biasanya nilai inputan akan diwakilkan pada karakter tertentu.

Contoh penggunaannya:

Password : <input type="password" name= "nmPword">

c. Input type = CheckBox

Elemen <input type=checkbox> merupakan elemen yang menyediakan pilihan yang bersifat boolean. Checkbox merupakan satu set elemen yang memiliki nama yang sama dengan nilai yang berbeda-beda. Pada checkbox disediakan n pilihan dari banyaknya pilihan yang disediakan.

Atribut yang diperlukan adalah:

- name:
  menunjukan kata yang bersifat umum, sesuai dengan pilihan yang disediakan.
- Value:
  adalah nilai dari checkbox yang akan dikirimkan melalui url.
- Checked:
  berhubungan dengan value (nilai) awal yang akan diambil dan bersifat optional.

Contoh penggunaannya:

Hoby: <input type="checkbox" name="hobi" value=" konsepweb" checked> Konsep Web

<input type="checkbox" name="hobi" value=" desainweb"> Desain Web

d. Input type = radio .

Sebuah elemen <input type=radio> menyediakan pilihan yang bersifat boolean dan merupakan satu set elemen yang memiliki nama yang sama dengan nilai yang berbeda-beda. Pada radio, hanya diper-kenankan untuk memilih 1 pilihan dari banyaknya

pilihan yang disediakan, akan tetapi type radio sangat ideal digunakan jika hanya memuat dua pilihan.

Atribut yang digunakan pada type radio sama dengan type checkbox.

Contoh penggunaannya:

Jenis Kelamin : <input type="radio" name="lakilaki" value="lakilaki" checked> Laki-laki <input type="radio" name="wanita" value="wanita"> Wanita.

e. Input type = image

Sebuah elemen <input type=image> menentukan sumber gambar yang akan ditampilkan dan memungkinkan untuk memasukan koordinat dari dua kolom formulir: x dan y dari pixel yang dipilih pada gambar. Nama-nama kolom adalah nama field dengan x dan y.

Atribut yang digunnakan adalah:

- name:
  diperlukan sebagai masukan untuk bidang lainnya.
- src:
  diperlukan untuk menjadi path pencarian file gambar yang akan ditampilkan.
- Align:
  untuk menentukan letak gambar pada layar dan bersifat optional.

Contoh Penggunaannya:

Pilih Titik : <input type=image name=point src="map.gif">

f. Input type = hidden

Sebuah elemen <input type=hidden> digunakan untuk memasukkan nilai pada form dimana value (data) inputannya tidak akan ditampilkan pada

layar. Tetapi, data tersebut tetap ikut dalam pengiriman isi form.

Contoh Penggunaannya:

<input type=hidden name="context" value="datatersembunyi">

g. Input type = submit

Sebuah elemen <input type=submit> merupakan pilihan input, yang biasa ditampilkan dalam bentuk tombol. Tombol tersebut akan menginstruksikan kepada browser untuk mengirimkan data-data yang ada di dalam form.

Atribut yang digunakan pada elemen ini adalah :

- Name:
  nama yang menunjukan persetujuan untuk mengirimkan data dalam form
- Value:
  adalah nilai yang akan ditampilkan pada tombol.

Contoh Penggunaannya:

<input type=submit value="OK">

h. Input type = reset

Sebuah elemen <input type=reset> merupakan pilihan input yang tampilannya hampir sama dengan type submit karena biasanya ditampilkan dalam bentuk tombol. Tombol tersebut akan menginstruksikan kepada browser untuk mereset atau menghapus semua data-data yang telah dimasukkan kedalam form.

Atribut yang digunakan pada elemen ini adalah:

- Name:
  nama yang menunjukan persetujuan untuk menghapus data dalam form.

- Value:
  adalah nilai yang akan ditampilkan pada tombol.
  Contoh Penggunaannya:
  <input type=reset value="Reset">

3. Elemen List

Html memiliki beberapa elemen untuk membut sebuah list daftar, yaitu:

a. Unordered List : UL, LI

Elemen <ul> merupakan elemen html yang digunakan untuk membuat daftar item yang biasanya memakai simbol bullet untuk list. Isi (elemen) dalam elemen <ul> adalah elemen <li> yang memungkinkan untuk membuat daftar yang bersarang.

Contoh penggunaan elemen <ul>:

```
<ul>
<li> Elemen 1
   <ul>
         <li> Elemen 1 dalam elemen 1</li>
         <li> Elemen 2 dalam elemen 1</li>
   </ul>
</li>
<li> Elemen 2</li>
<li> Elemen 3</li>
</ul>
```

b. Ordered List : OL

Elemen <ol> merupakan elemen yang membuat daftar (list) dimana penomoran pada item menggunakan daftar bernomor. Sama seperti UL, di dalam elemen OL juga memuat elemen <li>. Sedangkan dalam elemen <li> sendiri, biasa memuat elemen-elemen lain seperti paragraf

( ). Hal ini mengindikasikan bahwa OL memungkin untuk membuat daftar yang bersarang.

Contoh:

<ol>

* Elemen 1*

* Ini adalah Paragraf dalam elemen < li > *

*
*

* Elemen 2 *

*
*

c. Direktory List : DIR

Elemen <dir> mirip dengan elemen <ul>. Ini merupakan list item pendek, biasanya sampai 20 karakter. Item dalam list direktori dapat diatur dalam beberapa kolom, biasanya lebarnya adalah 24 karakter.

Isi dari elemen <dir> adalah urutan elemen <li>. Blok elemen bersarang tidak diperbolehkan dalam isi elemen <dir>.

Contoh :

<dir>

* AH <li> IM*

* MR <li> SZ*

</dir>

d. Menu List : MENU

Elemen <MENU> adalah daftar item yang biasanya satu baris per item. Gaya daftar menu biasanya lebih kompak dari gaya unordered list. Isi dari elemen <MENU> adalah urutan elemen <li>. Blok elemen bersarang tidak diperbolehkan dalam isi elemen <MENU>.

Contoh:

<menu>

* Pertama item dalam daftar.*

* Kedua item dalam daftar.*

* Ketiga item dalam daftar.*

</menu>

e. Definisi List: DL, DT, DD

Sebuah daftar definisi adalah daftar istilah dan definisi yang sesuai. Daftar definisi biasanya diformat dengan istilah flush-kiri. Isi dari elemen <dl> adalah urutan elemen <dt> dan/atau elemen <dd> dan berpasangan. Beberapa elemen <dt> dapat dipasangkan dengan elemen <dd> tunggal. Dokumen tidak boleh mengandung beberapa elemen <dd> secara berturut-turut.

Contoh penggunaan:

<dl>

<dt> <dd> INI adalah definisi dari istilah pertama.

<dt> <dd> ITU adalah definisi dari istilah kedua.

</dl>

4. Frase

Elemen frase mungkin bersarang dalam konten dari elemen frase lain, namun browser sendiri dapat membuat elemen frase bersarang. Elemen-elemen yang berkaitan dengan frasa dalam aturan html adalah:

Citation: CITE

Elemen <CITE> digunakan untuk menunjukkan judul buku atau kutipan lain. Hal ini biasanya diberikan sebagai huruf miring.

Sebagai contoh:

Buku Panduan Web adalah <cite> Konsep dan Desain Web </cite>

Kode : CODE

Elemen <code> menunjukkan contoh kode, biasanya diberikan dalam font mono-spasi. Elemen <code> ditujukan untuk kata yang pendek kata atau frase kode.

Sebagai contoh:

Ekspresi <code> x + = 1 </code>

adalah singkatan <code> x = x + 1 </code>

Penekanan : EM

Elemen menunjukkan sebuah penekann pada frase biasanya diberikan sebagai huruf miring.

Sebagai contoh:

Sebuah subjek tunggal selalu membutuhkan kata kerja tunggal

Keyboard : KBD

Elemen <KBD> mengindikasikan teks diketik oleh pengguna, biasanya diberikan dalam font mono-spasi. Ini umumnya digunakan dalam instruksi manual.

Sebagai contoh:

Masukkan <kbd> MENCARI TI </kbd> untuk mencari database

Contoh : SAMP

Unsur <SAMP> menunjukkan urutan karakter literal, biasanya diberikan dalam font mono-spasi.

Sebagai contoh:

Salah satu nama yang memuat <samp> andr </samp> adalah Andrey

Penekanan Kuat : STRONG

Elemen menunjukkan penekanan kuat, biasanya diberikan dalam huruf tebal.

Sebagai contoh:

Dilarang MEROKOK </ strong> ketika anda memasuki kampus "UIN Alauddin Makassar </ strong>".

Variabel : VAR

Elemen <VAR> menunjukkan variabel placeholder, biasanya diberikan sebagai huruf miring.

Contoh:

Jenis <SAMP> berkas untuk <VAR> memeriksa berkas html </VAR> | </SAMP> untuk memeriksa file <VAR> </VAR> untuk kesalahan markup.

Bold : B

Elemen <B> menunjukkan teks tebal.

Contoh untuk menampilkan **HURUF TEBAL** maka penulisan scriptnya adalah *HURUF TEBAL*

Huruf Miring : I

Elemen <I> menunjukkan teks miring.

Contoh untuk menampilkan *HURUF MIRING* maka penulisan scriptnya adalah *HURUF MIRING*

Anchor : A

Elemen <a> menunjukkan anchor hyperlink. Didalam anchor memuat beberapa atribut penting salah satunya adalah NAMA dan atribut HREF.

Atribut dari elemen <A> adalah:

- HREF
  Memberikan URI dari header hyperlink pada anchor.
- NAMA
  Memberikan nama anchor, dan membuatnya tersedia sebagai header dari hyperlink.
- TITLE
  Menunjukkan judul untuk sumber daya tujuan.

- REL
  Atribut REL memberikan hubungan yang dijelaskan oleh hyperlink.
- REV
  Sama dengan atribut REL, tetapi semantik dari hubungan berada di arah sebaliknya. Sebuah link dari A ke B dengan REL = "X" mengungkapkan hubungan yang sama sebagai link dari B ke A dengan REV = "X".

Line Break : BR

Elemen <BR> menentukan satu baris antara kata.

Sebagai contoh:

*<P> Pada penulisan ini akan di buat
*

Baris barus pertama
 Baris baru kedua
Baris Baru ketiga.

Horizontal Rule: HR

Unsur <HR> adalah pembatas antara bagian teks, biasanya sebuah lebar penuh aturan horizontal atau grafis setara.

Contoh:

<HR> <alamat> 8 Februari 1995, CERN </alamat>

5. **Image : IMG**

Unsur <img> mengacu pada gambar atau ikon melalui hyperlink. Atribut yang terdapat pada elemen IMG meliputi:

- ALT
  Atribut ALT akan diterjemahkan oleh browser sebagai sebuah nilai apabila file yang didefenisikan pada SRC tidak ditemukan.
- SRC
  Atribut SRC merupakan URI yang menunjukan lokasi dimana file image ditempatkan.

- ALIGN

  Merupakan penempatan gambar terhadap garis pinggir pada sebuah tabel atau field. Align dapat berupa:

  a. TOP menetapkan bahwa atas gambar sejalan dengan item tertinggi pada baris yang berisi gambar.
  b. TENGAH menetapkan bahwa tengah gambar sejalan dengan dasar dari baris yang berisi gambar.
  c. BOTTOM menetapkan bahwa bagian bawah gambar sejalan dengan dasar dari baris yang berisi gambar.

- ISMAP menunjukkan peta gambar.

  Contoh penggunaan:

  <IMG SRC="triangle.xbm" ALT="Warning:"> Pastikan untuk membaca petunjuk ini.

  **

  <IMG SRC="contoh.jpg">

  **

1.6 HyperText Transfer Protocol (HTTP)

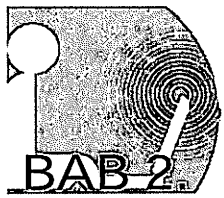
HTTP adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif dan menggunakan hipermedia. Penggunaannya banyak pada pengambilan sumber daya yang saling terhubung dengan tautan, yang disebut dengan dokumen hiperteks. Ada dua versi mayor dari protokol HTTP, yakni HTTP/1.0 yang menggunakan koneksi terpisah untuk setiap dokumen dan HTTP/1.1 yang dapat menggunakan koneksi yang sama untuk melakukan transaksi. Dengan demikian, HTTP/1.1 bisa lebih cepat karena memang tidak perlu membuang waktu untuk pembuatan koneksi berulang-ulang.

Pengembangan standar HTTP telah dilaksanakan oleh Konsorsium World Wide Web (World Wide Web Consortium/ W3C) dan juga Internet Engineering Task Force (IETF), yang berujung pada publikasi beberapa dokumen Request for Comments (RFC), dan yang paling banyak dirujuk adalah RFC 2616 (yang dipublikasikan pada bulan Juni 1999), yang mendefinisikan HTTP/1.1. Dukungan untuk HTTP/1.1 yang belum disahkan, yang pada waktu itu RFC 2068, secara cepat diadopsi oleh banyak pengembang penjelajah web pada tahun 1996 awal. Hingga maret 1996, HTTP/1.1 yang belum disahkan itu didukung oleh Netscape 2.0, Netscape Navigator Gold 2.01, Mosaic 2.7, Lynx 2.5, dan dalam Microsoft Internet Explorer 3.0. Pada bulan Maret 2006, salah satu perusahaan Web hosting melaporkan bahwa lebih dari 40% dari penjelajah Web yang digunakan di Internet adalah penjelajah Web yang mendukung HTTP/1.1. Perusahaan yang sama juga melaporkan bahwa hingga Juni 1996, 65% dari semua penjelajah yang mengakses *server*-server mereka merupakan penjelajah Web yang mendukung HTTP/1.1. Standar HTTP/1.1 yang didefinisikan dalam RFC 2068 secara resmi dirilis pada bulan Januari 1997. Peningkatan dan pembaruan terhadap standar HTTP/1.1 dirilis dengan dokumen RFC 2616 pada bulan Juni 1999. HTTP versi 1.1 dikeluarkan untuk mengakomodasi *proxy, cache* dan koneksi yang persisten.

HTTP merupakan protokol request/response antara client dan server. Sebuah klien HTTP (seperti *web browser*), biasanya memulai permintaan layanan dengan membuat hubungan ke port tertentu pada sebuah Web Server (biasanya *port* 80). Klien yang mengirimkan permintaan layanan tersebut dikenal dengan nama ***user agent***. Sementara untuk server yang merespons layanan dan menyimpan sumber daya seperti berkas HTML dan gambar, dikenal juga sebagai ***origin server***. Antara user agent dan *origin server* tersebut ada beberapa teknologi penghubung, seperti *proxy, gateway,* dan

juga *tunnel*. Pengaksesan sumber daya dengan menggunakan Uniform Resource Identifier (URI) atau lebih khusus melalui Uniform Resource Locator (URL) dengan menggunakan skema URI http: atau https:.

Penggunaan HTTP tidak terbatas pada TCP/IP, meskipun HTTP merupakan salah satu protokol aplikasi TCP/IP paling populer melalui Internet. HTTP dapat diimplementasikan di atas protokol yang lain di atas Internet atau di atas jaringan lainnya, akan tetapi protokol HTTP membutuhkan sebuah protokol lapisan transport yang dapat diandalkan.{"}



BAB 2 WEB Klien

Pada pembahasan web klien, akan dipaparkan bahwa terdapat beberapa hal penting yang berkaitan dengan web klien yakni:

- Peramban Web (*Web Browser*)
- Perkembangan Teknologi Web
- Fitur dan Fungsi Pada *Web Browser*
- *Web Browser* dan *HTTP Request*
- Browser Caching
- Konfigurasi *Browser*

2.1 Peramban Web

Peramban web (Inggris: *web browser*) disebut juga *penjelajah web* adalah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh ***web server***. Meskipun penggunaan *browser* banyak ditujukan untuk mengakses *World Wide Web* akan tetapi *browser* juga dapat digunakan untuk mengakses informasi yang diberikan oleh *web server* dalam jaringan pribadi atau file dalam sistem file. *Web browser* menyediakan antarmuka dengan pengguna berbasis teks dengan dukungan untuk HTML yang kaya dengan user interface untuk berbagai format file dan protokol. Web browser juga memiliki fitur tambahan untuk mendukung *e-mail* , *Usenet news* dan *Internet Relay Chat*(IRC).

Secara umum *web browser* memungkinkan pengguna untuk membuka beberapa sumber informasi pada saat yang sama, baik dalam jendela browser yang berbeda atau di berbagai tab dari jendela yang sama. Pada *web browser* juga terdapat *pop-up blocker* yang dapat digunakan untuk mencegah jendela yang tidak diinginkan muncul tanpa persetujuan pengguna. Kebanyakan *web browser* dapat menampilkan daftar halaman web yang telah dimasukkan oleh pengguna kedalam *bookmark* sehingga pengguna dapat dengan cepat kembali mendapatkan halaman tersebut. *Bookmark* juga disebut "Favorit" di Internet Explorer . Selain itu, hampir semua web browser memiliki beberapa bentuk *built-in web feed aggregator*. Pada Firefox, web feed yang diformat sebagai Live bookmark hidup dan berperilaku seperti folder penanda yang sesuai dengan entri terbaru di feed. Sementara Opera menyediakan pembaca feed yang lebih tradisional termasuk dalam menyimpan dan menampilkan isi feed. Selain itu, kebanyakan browser mendukung untuk peningkatan kemampuan browser melalui instalasi plug-in.

Browser web yang paling besar memiliki elemen-elemen antarmuka pengguna yang sama, yaitu:

- Tombol Kembali *(back)* dan tombol maju *(forward)* untuk kembali ke sumber daya sebelumnya.
- Sebuah tombol *refresh* atau *reload* untuk reload sumber daya saat ini.
- Sebuah tombol *berhenti* untuk membatalkan pemuatan sumber daya. Pada beberapa browser, tombol stop yang tergabung dengan tombol reload.
- Sebuah tombol *home* untuk kembali ke halaman awal pengguna.
- Sebuah *address bar* untuk memasukkan *Uniform Resource Identifier* (URI) dari sumber daya yang diinginkan dan menampilkannya.

- Sebuah bar pencarian memasukan kata kunci ke dalam mesin pencari(*search engine*). Pada beberapa browser, bar pencarian bergabung dengan address bar.
- Sebuah status bar untuk menampilkan proses dalam pemuatan sumber daya dan juga link URI ketika kursor melayang di atasnya dan kemampuan zoom (untuk memperbesar atau memperkecil ukuran layar).

Web Browser dapat dibagi dalam beberapa jenis, yaitu:

1. *Web browser for PC,* artinya browser yang bisa diinstalkan dan digunakan pada komputer PC/Laptop.
2. *Web Browser for Mobile,* artinya browser yang khusus digunakan pada perangkat mobile seperti : PDA, HP dan sejenisnya.

Berdasarkan sistem operasi untuk menjalankan browser, makan web browser PC dapat dikategorikan menjadi :

1. *Web Browser for Windows* yaitu, web browser yang bisa di instal dan digunakan pada OS Windows
2. *Web Browser for Linux* yaitu, web browser yang bisa di install dan digunakan pada OS Linux
3. *Web Browser for Mac* yaitu, web browser yang bisa di install dan digunakan pada OS Machintos

2.2 Perkembangan Browser

Peramban web pertama diciptakan pada tahun 1990 oleh Sir Tim Berners Lee yang disebut WorldWideWeb(tanpa spasi) dan kemudian berganti nama menjadi Nexus. Pada tahun 1993, perangkat lunak browser berbasis grafis mulai banyak digunakan, setelah seorang mahasiswa bernama **Marc Andressen** di University of Illinois di Urbana-Champaign- Amerika Serikat, membuat sebuah peramban web berbasis grafis pertama yang berjalan di atas sistem operasi Windows dan UNIX (berbasis Motif). Peramban web

tersebut dinamai **Mosaic**. Selanjutnya, setelah lulus dari universitas, Marc ditawari oleh Jim Clarc, salah seorang petinggi *Silicon Graphics Incorporated*(SGI), untuk membuat perusahaan dengan nama Mosaic Communication yang kemudian berubah menjadi Netscape Communication. Marc membuat sebuah peramban web populer pertama yang digunakan oleh umum, yang disebut dengan Netscape Navigator.

Pengenalan Mosaic pada tahun 1993 dan merupakan salah satu web browser grafis pertama, menyebabkan ledakan dalam menggunakan web. Andreessen, pemimpin tim Musa di NCSA, segera mendirikan perusahaan sendiri bernama Netscape dan merilis Mosaic. Pada tahun 1995, Microsoft dipengaruhi Netscape Navigator, yang dengan cepat menjadi browser paling populer di dunia, mengeluarkan produk mereka dengan nama Internet Explorer. Berada dalam sistem operasi Windows, membuat Internet Explorer memperoleh dominasi di pasar browser web. Internet Explorer berbagi penggunaan memuncak di lebih dari 95% pada tahun 2002. Pada tahun 1996 Opera diluncurkan, meskipun belum pernah gunakan secara luas akan tetapi penggunaan browser secara keseluruhan terfokus dan cepat tumbuh pada pasar ponsel web dan terinstal pada lebih dari 40 juta ponsel. Pada tahun 1998, Netscape diluncurkan dan berubah nama menjadi Mozilla Foundation dalam upaya untuk menghasilkan browser kompetitif dengan menggunakan model perangkat lunak yang open source. Pada Januari 2003, dirilis versi beta pertama untuk browwer Safari dan pada April 2011. Chrome sebagai pendatang baru di dunia browser pertama kali dirilis pada bulan September 2008. Chrome take-up telah meningkat secara signifikan dari tahun ke tahun, dengan menggandakan pangsa penggunaannya dari 8% menjadi 16% pada bulan Agustus 2011.

2.3 Fitur dan Fungsi pada Web Browser

Penting untuk memahami fungsi dan fitur dari web browser yang Anda gunakan. Pengaktifan beberapa fitur web browser terkadang berpengaruh dengan segi keamanan pada komputer pengguna. Tetapi, biasanya vendor akan mengaktifkan beberapa fitur secara default untuk meningkatkan kinerja pada komputer pengguna. Penyerangan pada sisi klien oleh hacker dan cracker biasanya terfokus pada bagaimana memanfaatkan sistem komputer melalui berbagai kelemahannya. Mereka menggunakan kerentanan ini untuk mengambil alih komputer, mencuri informasi, dan menghancurkan file serta menggunakan komputer pengguna untuk menyerang komputer lain. Sebuah cara mudah untuk melakukan penyerangan kepada pengguna adalah dengan memanfaatkan kerentanan dalam *web browser*. Seorang penyerang dapat membuat halaman web yang berbahaya yang akan menginstal *Trojan* atau spyware yang akan mencuri informasi pribadi pengguna. Pengetahuan tentang fungsi dan fitur pada web browser membantu pengguna memahami bagaimana fungsi web browser berpengaruh pada keamanan komputer.

Beberapa fitur utama yang terdapat pada web browser adalah sebagai berikut:

2.3.1 ActiveX

ActiveX adalah teknologi yang digunakan oleh Microsoft Internet Explorer pada sistem Microsoft Windows. ActiveX memungkinkan aplikasi atau bagian dari aplikasi yang akan digunakan oleh web browser. Sebuah halaman web dapat menggunakan komponen ActiveX yang mungkin sudah berada pada sistem Windows, atau sebuah situs dapat menyediakan komponen sebagai objek download. Kontrol ActiveX adalah program kecil, kadang-kadang disebut *add-ons* yang digunakan di Internet. Mereka dapat meningkatkan

kemampuan browsing dengan memungkinkan animasi atau membantu tugas-tugas seperti menginstal update keamanan pada Microsoft Update. Pengguna sebenarnya diberikan pilihan apakah mereka ingin mendownload kontrol ActiveX berdasarkan kasus per kasus. Beberapa situs mengharuskan pengguna untuk menginstal kontrol ActiveX demi melihat atau melakukan tugas tertentu pada website mereka. Biasanya, ketika pengguna mengunjungi situs yang menggunakan kontrol ActiveX, Internet Explorer (dalam hal ini sebagai web browser) akan bertanya apakah pengguna ingin menginstal kontrol ActiveX atau tidak.

2.3.2 Applet Java

Sebuah applet adalah tipe yang spesial dari program java yang dieksekusi melalui internet. Secara khusus berjalan pada suatu web browser seperti Netscape Navigator, Mozilla atau Microsoft Internet Explorer. Untuk alasan keamanan biasanya applet cukup terbatas jika dibandingkan dengan aplikasi yang berbasis Java. Applet Java merupakan bahasa pemrograman berorientasi objek yang dapat digunakan untuk mengembangkan konten aktif di situs web. Sebuah **applet Java** adalah applet dikirim ke pengguna dalam bentuk Java bytecode. Applet Java dapat dijalankan dalam web browser menggunakan Java Virtual Machine(JVM) atau di Sun's appletviewer. Applet Java diperkenalkan pada versi pertama dari bahasa Java pada tahun 1995 dan ditulis dalam bahasa pemrograman yang mengkompilasi ke Java bytecode, biasanya di Java tetapi dapat juga dalam bahasa pemrograman lain seperti Jython , JRuby atau Eiffel (via SmartEiffel).

Keuntungan Penggunaan Applet Java adalah sebagai berikut:

- Didukung oleh sebagian besar browser dan multi platform baik pada sistem operasi Linux, Microsoft Windows maupun Mac.

- Java applet meningkatkan kemampuan browser cache untuk memuat kembali suatu halaman web yang pernah diakses.
- Dapat memindahkan pekerjaan dari Server ke klien, membuat solusi web lebih terukur dengan jumlah pengguna/klien.

Kekurangan menggunakan Applet Java:

- Membutuuhkan Java plug-in .
- Beberapa browser terutama aplikasi browser pada perangkat seluler, Apple iOS atau Android tidak mendukung applet java.
- Seperti halnya pada scripting client side, pembatasan keamanan dapat membuat sulit atau bahkan tidak memungkinkan bagi sebuah applet untuk berjalan secara baik sehingga sulit untuk mencapai tujuan yang diinginkan oleh aplikasi.
- Beberapa applet membutuhkan JRE tertentu.
- Tidak ada standar untuk membuat isi dari applet yang tersedia bagi pembaca layar. Oleh karena itu, applet dapat membahayakan aksesibilitas dari sebuah situs web untuk pengguna dengan kebutuhan khusus.

2.3.3 Plug-in

Plug-in sebenarnya mirip dengan kontrol ActiveX, tetapi tidak dapat dijalankan di luar web browser. Plug-in dapat mengandung kelemahan pemrograman seperti buffer overflows atau plug-in mungkin mengandung cacat desain seperti pelanggaran lintas domain. Dalam komputasi *plug-in* adalah seperangkat komponen perangkat lunak yang menambahkan kemampuan khusus yang lebih besar pada perangkat lunak aplikasi. Kemampuan yang dimaksud dapat berupa kemampuan untuk memutar video, memindai virus dan

menampilkan jenis file baru. Beberapa plug-in yang terkenal antara lain Adobe Flash Player , QuickTime dan Java Applet. Add-on dalam komputasi sering dianggap sebagai istilah umum yang terdiri dari snap-in , plug-in, ekstensi, dan tema untuk aplikasi perangkat lunak.

Beberapa alasan utama penggunaan plug-in meliputi:

- Memungkinkan pengembang pihak ketiga untuk menciptakan dan memperpanjang kemampuan aplikasi
- Mendukung untuk menambahkan fitur baru pada browser
- Untuk mengurangi ukuran sebuah aplikasi
- Untuk memisahkan kode sumber dari aplikasi karena tidak sesuai lisensi perangkat lunak.

2.3.4 Cookie

Cookie adalah file yang ditempatkan pada sistem klien yang digunakan untuk menyimpan data bagi situs web tertentu. Cookies dapat berisi informasi tentang situs yang Anda kunjungi atau bahkan mungkin berisi mandat untuk mengakses situs. Cookie dirancang untuk dapat dibaca hanya oleh situs web yang membuat cookie itu sendiri. Session cookies dibersihkan bila browser ditutup dan persistent cookies akan tetap pada komputer sampai tanggal kedaluwarsa yang ditetapkan tercapai.

Cookie dapat digunakan secara unik untuk mengidentifikasi pengunjung situs web dan terkadang hal ini dipandang oleh sebagian orang sebagai pelanggaran privasi. Jika sebuah situs web menggunakan cookie untuk otentikasi maka penyerang dapat memperoleh akses tidak sah ke situs tersebut dengan mendapatkan cookie. Persistent cookies menimbulkan risiko lebih tinggi dari session cookies karena mereka tetap pada komputer lama.

2.3.5 JavaScript

JavaScript yang juga dikenal sebagai ECMAScript, adalah bahasa scripting yang digunakan untuk membuat situs web lebih interaktif. Ada spesifikasi dalam standar JavaScript yang membatasi fitur tertentu seperti mengakses file lokal.

2.3.6 VBScript

VBScript merupakan bahasa scripting yang unik untuk Microsoft Windows Internet Explorer. VBScript serupa dengan JavaScript, tetapi tidak banyak digunakan di situs web karena kompatibilitas terbatas dengan browser lainnya.

2.3.7 Live Bookmark

Live Bookmarks adalah *internet bookmark,* khususnya di Mozilla Firefox yang didukung oleh RSS . Ini memungkinkan pengguna untuk secara dinamis memantau perubahan sumber-sumber berita favorit mereka. Live Bookmarks akan diperbarui secara otomatis, namun proses update otomatis pada live bookmark dapat bersifat optional sehingga pengguna dapat melakukan konfigurasi khusus pada web browser.

Kemampuan untuk menjalankan bahasa scripting seperti JavaScript ataupun VBScript memungkinkan halaman untuk menambahkan sejumlah besar fitur dan interaktivitas ke halaman web. Namun, kemampuan yang sama dapat disalahgunakan oleh penyerang. Penyalahgunaan fungsi-fungsi tersebut dapat diminimalisir dengan cara melakukan konfigurasi pada browser.

2.4 Browser Request

Setiap kali browser meminta sebuah file (halaman, gambar dan lain-lain) dari web server maka protokol yang digunakan adalah HTTP. Sebagaimana yang dipaparkan pada bahasan sebelumnya pada bab I, bahwa *Hypertext*

Transfer Protocol adalah protokol request/response, yang berarti komputer klien mengirimkan permintaan untuk mendapatkan file (misalnya "berikan saya file 'home.html'") dan web server mengirimkan kembali respon ("Ini filenya", diikuti oleh file itu sendiri). Pada *browser request* setidaknya ada dua hal utama yang akan dibahas yaitu *The Request Line (Baca :* **Metode Pada HTTP)** dan *Request Header Field Lines.* Untuk lebih jelasnya mengenai proses *request* dari komputer klien, maka dalam buku ini akan diangkat sebuah contoh permintaan seperti yang tertulis di bawah ini.

Permintaan HTTP berikut ini diterima dari alamat IP 74.125.16.81 (port 63676) berdasarkan alamat IP 81.187.237.83 (port 80):

```
GET /dumprequest HTTP/1.1
Host: djce.org.uk
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Referer:
http://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CEwQFjAD&
url=http%3A%2F%2Fdjce.org.uk%2Fdumprequest&ei=vEKWT538PMLorAf8p-
z2DQ&usg=AFQjCNEeAn5wSZMp_y_oTmOKonq482sS9A&sig2=_AHQJZn6wLI3jOBM3V
8J18A
```

Gambar 2.1 *Contoh HTTP Request*

Detail dari permintaan di atas adalah:

IP address Sumber, port, host dan protokol

IP address sumber	*: 74.125.16.81*
Port sumber	*: 63676*
Via	*: not present*
X-Forwarded-For	*: not present*

Untuk mengirim respon yang tepat kembali ke komputer klien, web server tentu mengetahui alamat IP dan nomor port yang mengirim respon. Alamat IP klien 74.125.16.81 dan nomor port yang digunakan adalah 63676. Di sisi lain, mungkin ada satu atau lebih *server proxy* antara komputer klien dengan web server. Jika permintaan HTTP termasuk header "Via" atau "X-Forwarded-For", maka itu mengindikasikan bahwa ada setidaknya satu proxy server. Jika tidak satupun dari header digunakan (not present) maka bisa berarti bahwa tidak ada proxy server yang terlibat atau bisa berarti bahwa server "menyembunyikan diri" dari klien. Header *X-Forwarded-For* biasanya dapat memberikan informasi kepada pengguna mengenai IP yang digunakan.

Alamat IP

Dari contoh di atas, kita dapat memastikan bahwa alamat IP klien adalah 74.125.16.81. Selanjutnya apa yang bisa kita ketahui tentang alamat tersebut?

Catatan:

Bagian ini tidak ada hubungannya dengan HTTP pada khususnya, ini hanya sebuah contoh informasi yang bisa didapat dari alamat IP

IP address : 222.124.152.177
DNS name : 177.subnet222-124-152.static.astinet.telkom.net.id

Banyak informasi yang lebih menarik dapat dipelajari dari alamat IP Klien. Misalnya, keberadaan pengguna di internet, kota asal dan ISP yang digunakan.

IP address tujuan, port, host dan protokol

IP address Tujuan	*: 81.187.237.83*
Port tujuan	*: 80*
Host	*: djce.org.uk*
Protokol	*: INCLUDED*

Header di atas memberikan kejelasan tentang web server mana yang akan dihubungi. Mengingat banyaknya situs web

yang *"hosting"* pada server tunggal maka ketika permintaan diterima akan didefenisikan situs web mana yang akan diakses oleh pengguna. Sementara protokol yang digunakan selalu berupa "HTTP/1.1" atau "HTTP/1.0", bergantung pada *properti web browser* dari komputer pengguna dan *proxy* yang digunakan untuk melewatkan permintaan tersebut.

Requested URI

Requested URI : /dumprequest

Bersamaan dengan header 'host' dan nomor port tujuan di atas, maka permintaan tersebut menentukan dokumen mana yang harus diambil. Melihat nilai yang terkandung pada contoh di atas maka kita dapat menentukan bahwa URL dari dokumen yang diambil adalah *http://djce.org.uk/dumprequest.*

Metode dan Isi Permintaan

Request method : *GET*
Data : *none*

Metode permintaan biasanya berupa GET atau POST. Pada dasarnya jika klien mengisi dan mengirimkan formulir pada halaman web maka akan menghasilkan permintaan POST (atau mungkin GET), sedangkan jika klien melakukan klik pada link atau mengaktifkan fitur pada browser baik penanda (*bookmark*) atau favorit, maka metode permintaan akan selalu GET. Permintaan pada browser berupa POST ada dua jenis yaitu *multipart / form-data* yang digunakan ketika meng*upload* file ke *web server* dan *application / x-www-form-urlencoded* yang biasa digunakan secara umum untuk data yang diisikan pada form.

User Agent

User-Agent :
Mozilla/5.0 (Windows NT 6.1; rv:8.0.1) Gecko/20100101 Firefox/8.0.1

Accept :
*text/html, application/xhtml+xml, application/xml;q=0.9, */*;q=0.8*
Accept-Charset : ISO-8859-1,utf-8;q=0.7,;q=0.7*
Accept-Encoding : gzip, deflate
Accept-Language : en-us,en;q=0.5

Header dari **User-Agent** menginformasikan browser yang digunakan oleh klien. User-Agent berisi nama dan versi browser (misalnya Firefox 1.0.7), Sistem Operasi pada komputer klien dan versinya (misalnya Windows XP) dan informasi tambahan seperti yang "service pack" yang telah diinstal. **Accept** menjelaskan kemampuan yang dimiliki oleh browser serta menjelaskan tipe dokumen yang dapat ditangani misalnya pengguna dapat mengetahui apakah browser mereka dapat menangani gambar yang berekstensi png (grafis) atau tidak. **Accept-Charset** menggambarkan rangkaian karakter apa yang dapat diterima, sehingga server dengan mudah dapat menebak daerah dan bahasa apa yang digunakan oleh klien. Sebagai contoh, klien yang berada di Eropa Barat atau Amerika utara mungkin hanya memahami rangkaian karakter "iso-8859-1", "us-ascii" dan "utf-8", sedangkan rangkaian karakter "big5" akan menunjukkan bahwa klien kemungkinan besar berasal dari Cina. **Accept-Encoding** menjelaskan kemampuan browser dalam menangani transfer dokumen yang terkompresi. Sedangkan **Accept-Language** menjelaskan bahasa apa yang digunakan untuk menerima dokumen. Misalnya, jika header memberitahu kita bahwa preferensi klien adalah untuk "en-gb" diikuti dengan "en", itu berarti klien kemungkinan besar seorang warga Inggris berbahasa Inggris. Sementara untuk "pt-br" dapat diterjemahkan sebagai Portugis tetapi menggunakan bahasa Brasil.

Referring Page

Referer:http://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CEwQFjAD&url=http%3A%2F%2Fdjce.org.uk%2Fdumprequest&ei=vEKWT538PMLorAf8pz2DQ&usg=AFQjCNEeAn5wSZMp_y_oTmOKonq482sS9A&sig2=_AHQJZn6wU3jOBM3V8JI8A

Header *referer* pada dasarnya menginformasikan dokumen mana yang akan menjadi acuan. Jika klien mengikuti pranala (link) untuk mendapatkan halaman tersebut maka itulah yang akan menjadi URL dari halaman tersebut. Di sisi lain jika klien tidak mengikuti link mungkin klien hanya mengklik *bookmark* pada browser atau mungkin klien hanya mengetik alamat halaman ini langsung ke browser maka kemungkinan besar *referer* akan hilang.

Cookies

Setiap *respon* layanan dari web server baik berupa halaman, grafis dan lain-lain, maka pada kesempatan yang sama ia juga akan mengirimkan cookie untuk web browser. Cookies adalah potongan-potongan kecil informasi yang disimpan pada browser dan kemudian mengirimkan kembali ke web server yang sama setiap kali pengguna meminta dokumen. Jadi ada dua poin penting di sini pertama setiap cookie hanya dikirim kembali ke situs web yang sama sesuai dengan alamat asal dan kedua isi dari cookie (data yang dikandungnya) hanya dapat terdiri dari apa pun informasi web server yang sudah diketahui sebelumnya.

Connection Control

Connection : *keep-alive*

Keep-Alive : *not present*

Header ini digunakan untuk menyempurnakan lalu lintas jaringan antara klien dan web server. Header ini tidak memberikan banyak informasi, kecuali nilai tentang status koneksi yaitu *keep-alive* atau *not present.*

Cache Control

Pragma : *not present*
Cache-Control : *not present*
If-Modified-Since : *not present*

Header ini mengontrol caching dokumen. Cache control ini, dapat dideteksi dengan penggunaan tombol refersh ketika pengguna memuat ulang halaman (*reload*).

Authorisation

Username : *not present*

Jika pengguna telah login pada suatu situs web, maka nama pengguna dapat ditampilkan pada halaman tersebut dengan catatan bahwa ini hanya berlaku untuk situs web yang menggunakan otentikasi HTTP yang tepat. Biasanya pada jendela login pengguna mendapatkan tiga kesempatan untuk memasukkan username dan password, jika tidak sesuai maka pengguna akan melihat halaman yang mengatakan "*Authentication Required*". Akan tetapi hal ini juga tidak berlaku untuk situs web di mana login ditempatkan pada halaman yang terpisah. Ada juga kemungkinan untuk memasukkan username dan password dalam URL, misalnya pengguna mengetikkan http://user:password@www.example.com/. Jadi Dalam kasus ini, username akan muncul di halaman ini juga.

2.5 Browser Caching

Web browsing mendominasi Internet saat ini. Lebih dari dua pertiga dari lalu lintas di Internet saat ini dihasilkan oleh Web. Memeriksa kinerja transaksi internet merupakan salah satu cara produktif untuk meningkatkan kualitas layanan yang disampaikan melalui Internet. Dari proses transaksi inilah, *web caching* dapat memainkan peran penting dalam meningkatkan kualitas layanan bagi pengguna internet.

Browser Cache, kadang-kadang juga disebut sebagai folder file sementara Internet, berisi file dari situs web yang dikunjungi. Semua web browser modern menjaga file cache untuk satu alasan penting yaitu untuk menampilkan halaman web lebih cepat di waktu berikutnya apabila pengguna membuka lagi situs-situs web tersebut. Alasan nya karena file halaman web bukan diambil dari *web server online* tetapi diambil dari *browser cache* (di komputer klien) sehingga halaman web dapat tampil lebih cepat. File dari web cahce biasanya tersimpan pada hard disk atau RAM. Apabila file cache tidak diperlukan dari fungsi web browser, maka klien dapat menghapus browser cache tersebut. File-file dari cache browser ini bisa memuat seluruh halaman web, gambar, CSS, audio, video dan lain-lain. Besar kecilnya ukuran file cache akan berpengaruh pada banyaknya ruang hard disk yang akan terpakai untuk menyimpan file cache. File cache yang besar akan membuat halaman dan file dari web akan semakin cepat di ditampilkan di browser klien. Yang perlu di perhatikan, ketika menutup jendela browser maka dengan sendirinya browser melakukan operasi pemeliharaan file cache. Jika waktu untuk menutup browser memakan waktu lama, maka klien hendaknya mempertimbangkan untuk mengurangi dan mengatur kembali ukuran file cache browser.

Ada dua jenis web cache Web yaitu browser cache dan proxy cache. Browser cache berfungsi untuk menyimpan salinan dari semua halaman yang baru ditampilkan kedalam komputer pengguna dan biasanya salinan tersebut bisa digunakan kembali. Sedangkan proxy cache adalah perangkat jaringan yang dapat melakukan transaksi Web atas nama klien. *Proxy cache* biasanya menangani permintaan klien dengan memberikan salinan lokal konten sehingga menghindari terulangnya download konten dari sumbernya yang asli. Tetapi pembahasan dalam buku ini, lebih mendalam membahas web cache dibandingkan dengan proxy cache.

Prinsip desain asli dari arsitektur internet adalah model end-to-end. Dalam model ini jaringan adalah instrumen pasif yang dapat meneruskan paket ke tujuan yang ditentukan. Setiap paket yang dihasilkan oleh sebuah host diasumsikan untuk diteruskan ke tujuan yang ditangani dan setiap tanggapan atas datagram diasumsikan datang dari alamat tujuan. HTTP (*Hypertext Transfer Protocol*), dibangun pada model ini di mana Web klien mengambil alamat URL yang menyebabkan sesi TCP akan dibuka dengan host target yang ditentukan. Transaksi HTTP berikutnya mengidentifikasi data yang diminta pada host tujuan dan selanjutnya data ini kemudian dilewatkan kembali ke klien. Model pengiriman terbaik dinyatakan sebagai model pengiriman just-in-time, dimana data akan diteruskan ke klien sesuai permintaan.

Keuntungan dari model pengiriman end to end antara lain:

- Server konten dapat memodifikasi konten dan semua permintaan klien berikutnya diberikan dengan informasi terbaru sehingga data yang disampaikan merupakan data yang ter*update*.
- Server dapat melacak semua permintaan konten dan memungkinkan bagi penyedia konten untuk melacak konten tertentu yang diminta, identitas masing-masing klien serta seberapa sering setiap item konten direferensikan.
- Penyedia konten juga dapat membedakan bentuk model keamanan yang digunakan pada tiap-tiap klien serta dapat mengotentikasi klien sehingga memungkinkan untuk memberikan informasi khusus bagi klien tertentu.

Adapun kelemahan pada model *end to end* antara lain:

- Adanya kelemahan balancing.

  Hal ini dapat terjadi pada sebuah server penyedia konten populer yang selalu berada pada tekanan

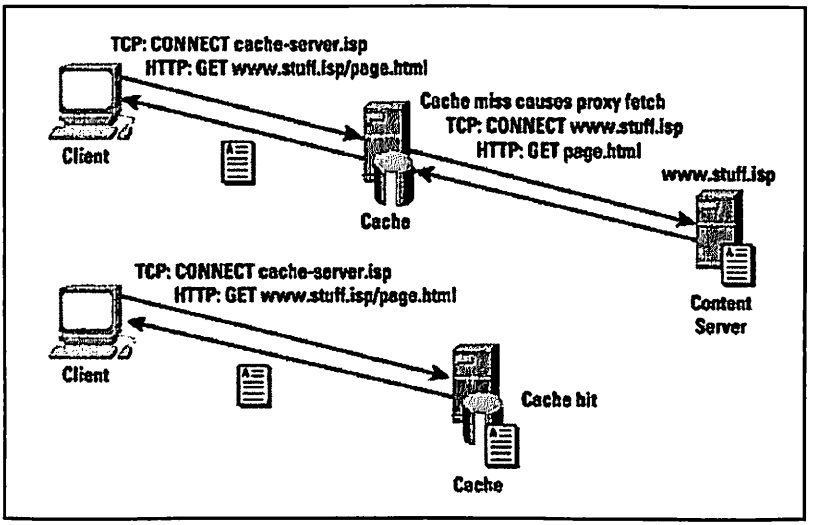
yang cukup besar, baik dalam jumlah koneksi klien simultan aktif setiap saat maupun pada volume total data yang dikirimkan dari server pada jaringan sekitarnya. Beban ini dinyatakan sebagai beban sistem server dan sebagai beban pada jaringan sekitarnya. Beberapa upaya meningkatkan kinerja sistem tersebut dapat dilakukan dengan meningkatkan jumlah server dan meningkatkan kapasitas dari jaringan lokal. Akan tetapi upaya tersebut mungkin tidak menjawab semua masalah dalam menjaga kualitas pengiriman konten. Sistem klien berbasis modem misalnya, dimana mereka menggunakan *low-bandwidth nirkabel* berbasis sistem klien yang dibatasi oleh kombinasi dari bandwidth yang terbatas dan berada pada hop terakhir sehingga menciptakan adanya keterlambatan dalam berinteraksi dengan server.

- Penggunaan jaringan yang tidak efisien.

  Lalu lintas Web memang memiliki tingkat duplikasi yang besar di mana satu set klien meminta salinan konten yang sama dan jaringan membawa duplikat data untuk setiap klien. Sementara untuk mentransmisikan data (mengimpor konten sekali saja) itu sendiri memerlukan biaya bisnis yang tidak sedikit.

Berdasarkan kelemahan di atas, maka penggunaan caching konten untuk penyedia konten dan pengguna akhir merupakan alternatif yang masuk akal. Hal ini dapat meningkatkan kinerja pelayanan pengiriman konten dalam jaringan serta meningkatkan efisiensi pengangkutan data dalam jaringan itu sendiri. Proses caching tersebut dimulai dari permintaan klien yang dilewatkan melalui agen cache dan membuat permintaan ke sumber aslinya sebagai proxy untuk klien. Respon dari server disimpan dalam cache lokal

dan salinannya akan diteruskan ke klien. Jika permintaannya sama maka akan diteruskan ke agen tembolok segera setelah permintaan asli dilayani. Selanjutnya respon dapat dihasilkan dari cache tanpa referensi lebih lanjut ke sumber asli. Pengoperasian Web cache ditunjukkan pada gambar di bawah ini.



Gambar 2.2 *Proses pada web cache*

Kecepatan Akses dan Browser Cashing

Pengukuran profil lalu lintas ISP menunjukkan bahwa sekitar 70 persen dari lalu lintas khas ISP adalah lalu lintas webbased. Analisis permintaan Web menunjukkan bahwa tingkat khas kesamaan permintaan (untuk objek yang sama seperti yang sebelumnya diminta) dapat mencapai 50 persen dari seluruh lalu lintas webbased. Proses peerhitungan tersebut dapat dilakukan dengan menghitung *hit rate*. Ada dua ukuran hit rate yaitu tingkat hit halaman dan sebuah byte hit rate. Sebuah hit rate halaman mengukur proporsi permintaan HTTP yang dapat disajikan dari cache, terlepas dari ukuran halaman. Sebuah byte hit rate mengukur rasio

jumlah byte yang dikirim dari cache hits terhadap jumlah byte yang meleset. Ukuran rata-rata transaksi Web adalah sekitar 16 paket data dalam aliran TCP. Dalam proses *slow-start* kontrol aliran TCP, siklus pertama akan mengirimkan satu paket dan menunggu ACK. Penerimaan ACK akan memicu transmisi dua paket lebih dalam siklus round-trip kedua, dan kemudian pengirim akan menunggu dua ACK. Penerimaan dari kedua ACK akan memicu empat paket lebih lanjut dalam siklus ketiga dan delapan dalam siklus berikutnya, dan paket tunggal yang tersisa dalam siklus kelima. Oleh karena itu, memungkinkan untuk perilaku yang optimal pada algoritma *slow-start* TCP, transaksi Web rata-rata membutuhkan beberapa round-trip. Jika pengguna berada agak jauh dari halaman Web dan waktu pulang-pergi ke sumber adalah 300 ms, delay propagasi dari beban halaman adalah 1,5 detik. Sebagai perbandingan, jika waktu pulang-pergi untuk cache Web lokal adalah 2 ms, maka delay propagasi dari beban halaman akan menjadi 10 ms. Angka-angka latency mengasumsikan jaringan *uncongested* dalam kedua kasus. Dalam hal ini, selama pencarian Web cache dapat menyelesaikan dalam waktu 1 detik, cache akan tampak jauh lebih cepat kepada pengguna. Uraian tersebut memberikan gambaran tentang manfaat untuk klien, dimana delay jaringan berkurang antara klien dan hasil cache lokal dalam peningkatan kecepatan pengiriman halaman Web untuk konten cache.

2.6 Konfigurasi Browser

Konfigurasi browser yang paling dalam lebih dari satu cara, metode pertama adalah yang paling sederhana untuk *sysadmin* dan yang kedua adalah konfigurasi bagi pengguna. Dalam buku ini konfigurasi untuk administrator system disebut sebagai *konfigurasi dasar* pertama sementara konfigurasi bagi pengguna diistilahkan dengan *konfigurasi lanjutan*.

2.6.1 Konfigurasi Dasar

Dalam mode ini, masing-masing browser dikonfigurasi secara independen. Jika *admin* mengubah sesuatu pada server (misalnya pada port yang menerima permintaan), maka masing-masing browser harus disetup ulang secara manual. Untuk menghindari caching situs intranet, *admin* harus menambahkan pengecualian bagi setiap situs. Untuk mengkonfigurasi browser apa pun, setidaknya memerlukan informasi mengenai:

- Nama host *server proxy*
- Port yang digunakan untuk menerima permintaan server proxy

Jika *admin* memutuskan untuk memindahkan cache ke komputer lain pada tahap berikutnya, maka akan ditemukan kenyataan bahwa mengubah pengaturan DNS itu jauh lebih mudah dibandingkan dengan mengubah konfigurasi dari setiap browser pada jaringan yang ada. Jika sistem operasi mendukung IP alias maka admin harus mengatur alamat IP khusus untuk *cache server* dan menggunakan pilihan *tcp_incoming_address* dan *tcp_outgoing_address* pada *squid.conf* untuk membuat Squid hanya menerima permintaan HTTP masuk pada alamat IP. Penamaan untuk cache bisa beragam, tapi pada umumnya menggunakan salah satu dari istilah berikut : *cache, proxy, www-proxy, www-cache* atau nama produk yang digunakan *cumi, NetApp ataupun netscape.*

Konfigurasi Dasar dapat memuat beberapa hal penting, antara lain:

- **Interaksi Browser-cache**

  Pada "tahap tertentu" server yang tidak benar menangani caching akan ditemukan, sehingga pengguna dapat dengan mudah menambahkan server yang tidak tepat ke dalam daftar tembolok. Namun kebanyakan browser memberikan akses pengguna untuk memaksa *reload* halaman web.

- **Pengujian Cache**

  Untuk menguji cache dengan benar maka dibutuhkan dua mesin setup guna mengakses cache dan halaman yang tidak mengandung header *"do not cache"*. Halaman yang menggunakan ASP sering termasuk header yang memaksa Squid untuk tidak melakukan cache pada halaman. Jadi, untuk menguji cache dapat dilakukan dengan memilih salah satu situs yang tidak aktif pada jaringan lokal (untuk mendapatkan perubahan yang nyata, memilih satu di negara yang berbeda) dan mengaksesnya dari mesin pertama. Setelah itu lakukan download dan pada mesin kedua lakukan download ulang halaman. Setelah halaman telah didownload, selanjutnya periksa apakah halaman tersebut ditandai sebagai 'HIT' (dalam file bernama *access.log*). Jika ada kesalahan pada penandaan akses yang kedua, maka hal itu disebabkan karena server asal meminta Squid untuk tidak melakukan cache pada halaman. Untuk melihat perbedaan cache dalam meningkatkan kecepatan browsing maka hal yang sama dapat dilakukan pada halaman yang berbeda. Akan tetapi jika situs yang dipilih terlalu dekat, maka akan ada kemungkinan hanya dapat melihat perbedaan kecepatan pada waktu transaksi (transaction-time) access.log tersebut.

- Cache Auto-config

  Browser Klien dapat memiliki semua pilihan konfigurasi secara manual, atau mereka dapat mengkonfigurasi untuk mendownload file autoconfig (setiap kali mereka start up), yang menyediakan semua informasi tentang setup cache. Setiap URL yang direferensikan (baik itu URL yang diketik maupun URL untuk grafik pada halaman belum diambil) akan diperiksa sesuai aturan yang ada. Oleh karena itu, pengguna diharapkan

memiliki aturan yang sesingkat mungkin jika tidak maka hal itu hanya akan memperlambat beban halaman, bukan pada tingkat cache melainkan pada browser.

- **Perubahan Konfigurasi Server Pada File Autoconfig**

  Dokumentasi untuk konfigurasi proxy pada Netscape menyarankan *proxy.pac* sebagai nama file untuk file autoconfig proxy. Oleh karena itu, jika ada sebuah file yang *berekstensi. Pac* yang tidak digunakan untuk autoconfiguration, maka browser memerlukan server untuk mengembalikan file autoconfig untuk menentukan type dari MIME file. Web Server tidak secara otomatis mengenali ekstensi *pac* sebagai file *proxy-autoconfig.*, dan harus mengkonfigurasi ulang untuk mengembalikan tipe mime ke keadaan yang benar *(application / x-ns-proxy-autoconfig).*

- **Super Proxy Script**

  ISP yang besar biasanya memiliki lebih dari satu *cache server.* Untuk menghindari duplikasi obyek maka server tersebut harus berkomunikasi antara satu dengan yang lain. Sementara untuk menemukan salinan objek, maka perlu dilakukan query antar cache. Contoh kasus pada situasi normal, cache1 mendapat permintaan untuk suatu objek. Maka proses permintaan halaman ini akan tersimpan pada disk. Satu jam kemudian, cache2 mendapat permintaan untuk objek yang sama. Akan tetapi, permintaan kedua ini diambil dari situs asli, sementara itu biasanya akan lebih cepat untuk mendapatkan tanggal dari cache1 yang terletak di dekatnya. Jika permintaan yang masuk untuk URL tertentu hanya ke satu cache, maka cache tidak perlu berkomunikasi dengan yang lain.

- **CGI yang Menghasilkan File Autoconfig**

  Hal ini dimungkinkan untuk mengasosiasikan *ekstensi. Pac* dengan program cgi di web server. Suatu program kemudian bisa menghasilkan skrip autoconfig tergantung

pada alamat dari sumber permintaan. Karena file autoconfig hanya dimuat pada startup (atau ketika tombol *refresh* ditekan) maka akan ada sedikit keterlambatan karena program cgi tidak akan terlihat oleh pengguna. Kebanyakan ISP besar mengalokasikan subnet ke berbagai wilayah, sehingga browser dapat dikonfigurasi guna mengakses cache terdekat sesuai dengan alamat sumber dari permintaan untuk program cgi.

2.6.2 Konfigurasi Lanjutan

Dalam mode ini, konfigurasi akan mengikuti aturan dari *server (server rule)*. Klien terhubung ke server pada startup dan mendownload informasi untuk berinteraksi dengan server proxy. Pengecualian dari situs tertentu akan ditangani secara terpusat, jadi satu perubahan akan memperbarui semua klien. Untuk antisipasi dari hal tersebut dapat dilakukan dengan menggunakan opsi auto-config. Meskipun metode ini disebut *auto-config*, namun tidak berarti semuanya berjalan secara otomatis, karena pengguna masih harus memasukkan URL yang menunjukkan lokasi dari daftar aturan.

Konfigurasi lanjutan memiliki beberapa keuntungan:

- Memudahkan Perubahan pada server proxy yaitu hanya dengan mengubah aturan server.
- Sebuah server proxy dapat dipilih berdasarkan nama mesin tujuan, port tujuan dan lain-lain.
- Memberi kemudahan pada konfigurasi browser
- Lebih cenderung menggunakan cache.

Konfigurasi Jendela Browser

Windows Klien

Lihat ***http://www.squid-cache.org/Doc/FAQ/FAQ-5.html*** untuk informasi lebih lanjut tentang konfigurasi jendela browser.

Unix klien

Kebanyakan Klien pada Unix menggunakan variabel tunggal untuk memutuskan bagaimana cara mengakses internet. Jika pengguna menjalankan lynx (browser berbasis terminal) pada setiap mesin mereka atau menggunakan web-spider wget rekursif, maka pengguna perlu menetapkan variable shell sehingga program ini berjalan pada server proxy yang benar. Setiap protokol memiliki variabel yang berbeda, sehingga pengguna dapat mengkonfigurasi browser untuk menggunakan proxy yang berbeda pada setiap protokol.

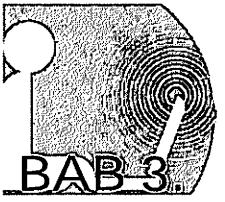
Konfigurasi jendela klien dapat memuat beberapa hal penting, antara lain:

- **Menonaktifkan popup window**
- **Menonaktifkan Fasilitas Gambar**
- **Mengatur langsung Konfigurasi (Browser Mozilla)**

Berikut cara untuk mengatur konfigurasi mozilla firefox di about:config:

- Buka Browser Mozilla anda dan ketikkan about:config lalu enter.
- Cari dan Ubah data-data seperti dibawah:
  - **network.http.pipelining**, ubah nilainya menjadi **true**
  - **network.http.proxy.pipelining**, ubah nilainya menjadi **true**
  - **network.http.pipelining.maxrequests** ubah nilainya menjadi **100** (isikan antara 30-100 semakin besar semakin cepat)
- Selanjutnya klik kanan pada area kosong untuk menampilkan popup menu. Pilih **New ==> Integer**, isikan dengan nama **nglayout.initialpaint.delay** dan isi nilainya dengan 0.{"}

Catatan:
Hati-hati ketika melakukan perubahan terhadap konfigurasi mozilla ini. Jangan mengubah data yang lain bila Anda tidak tahun kegunaannya, sebab akan dapat menyebabkan mozilla Anda tidak berjalan dengan sempurna. Dan bila hal itu terjadi, kembalikan saja ke nilai konfigurasi awal mozilla sebelum diubah atau instal ulang mozilla firefox.



BAB 3 WEB Server

Topik Utama pada bab ini, meliputi:

- Web Server
- *HTTP Request* dan *HTTP Response*
- Penanganan *HTTP Request* dan *HTTP Response*
- Arsitektur Web Server

3.1 Web Server

Tahun 1989, *Tim Berners-Lee* mengajukan pada perusahaannya CERN (*European Organization for Nuclear Research*) sebuah proyek yang bertujuan untuk mempermudah pertukaran informasi antar para peneliti dengan menggunakan sistem hiperteks. Sebagai hasil atas implementasi proyek ini, maka pada tahun 1990 *Berners-Lee* menulis dua program komputer:

- Sebuah peramban yang dinamainya sebagai WorldWideWeb;
- Web Server pertama di dunia, yang kemudian dikenal sebagai *CERN httpd* yang berjalan pada sistem operasi NeXTSTEP.

Dari tahun 1991 hingga 1994, kesederhanaan serta efektifitas atas teknologi yang digunakan untuk berkunjung serta bertukar data melalui *Waring Wera Wanua* membuat kedua aplikasi tersebut diadopsi pada sejumlah sistem operasi

agar dapat digunakan oleh lebih banyak individu ataupun kelompok. Awalnya adalah organisasi penelitian, kemudian berkembang dan digunakan di lingkungan pendidikan tinggi dan akhirnya digunakan dalam industri bisnis. Tahun 1994, Tim Berners-Lee memutuskan untuk membakukan organisasi *World Wide Web Consortium*(W3C) untuk mengatur pengembangan-pengembangan lanjut atas teknologi-teknologi terkait lainnya (HTTP, HTML dan lain-lain) melalui proses standarisasi.

Web Server dapat merujuk baik pada perangkat keras ataupun perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs web dalam layanan ke pengguna dengan menggunakan aplikasi tertentu seperti peramban web. Penggunaan paling umum Web Server adalah untuk menempatkan situs web, namun pada prakteknya penggunaannya diperluas sebagai tempat peyimpanan data ataupun untuk menjalankan sejumlah aplikasi kelas bisnis.

Saat ini umumnya *Web Server* telah dilengkapi pula dengan mesin penerjemah bahasa skrip yang memungkinkan *Web Server* menyediakan layanan situs web dinamis dengan memanfaatkan pustaka tambahan seperti PHP dan *Active Server Pages* (ASP). Hal ini mengindikasikan bahwa perilaku dari web server dapat ditulis dalam file terpisah, sedangkan perangkat lunak server yang sebenarnya tetap tidak berubah. Pada prakteknya ia juga dapat ditemukan tertanam (*embedded*) pada perangkat keras lain seperti printer, router, web cam yang menyediakan akses layanan http dalam jaringan lokal dan ditujukan untuk menyediakan manajemen perangkat serta mempermudah peninjauan atas perangkat keras tersebut. Hal ini berarti bahwa tidak ada perangkat lunak tambahan harus diinstal pada komputer klien, karena sebagian besar sistem operasi sekarang telah mendukung

untuk diinstalnya *web browser*. Fungsi utama sebuah Web Server adalah untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Pengguna biasanya melalui aplikasi pengguna seperti peramban web, meminta layanan atas berkas ataupun halaman web yang terdapat pada sebuah *web server*, kemudian server sebagai manajer layanan tersebut akan merespon balik dengan mengirimkan halaman dan berkas-berkas pendukung yang dibutuhkan atau menolak permintaan tersebut jika sumber daya yang diminta tidak tersedia.

Kemampuan yang terdapat pada web server didukung oleh beberapa fitur utama yang terdapat didalamnya, yaitu:

- *Virtual Hosting* untuk melayani banyak website dengan menggunakan satu IP
- *Large file support* untuk dapat melayani file besar yang kapasitasnya lebih besar dari 2 GB pada sistem operasi 32 bit.
- *Bandwidth Throttling* yang digunakan untuk dapat melayani lebih banyak klien dan membatasi respon agar tidak terjadi kejenuhan pada jaringan.
- *Server Side Scripting* untuk menghasilkan halaman website dinamis.

Untuk mempertahankan kemampuan yang terdapat pada web server, maka perlu dilihat beban kerja yang ditanggung oleh web server. Beban kerja tersebut mencakup batas beban kerja, penyebab kelebihan beban kerja dan gejala kelebihan beban kerja maupun teknik untuk memperbaikinya.

Batas Beban Kerja

Sebuah web server (program) telah menentukan batas beban, karena hanya dapat menangani sejumlah koneksi dari klien (biasanya antara 2 dan 80.000, secara default antara

500 dan 1.000) per alamat IP. Proses pengaturan jumlah beban kerja untuk permintaam maksimum per detik ditentukan oleh:

- Pengaturan web server sendiri
- Jenis permintaan HTTP
- Tipe halaman (*content*), apakah yang statis atau dinamis
- Apakah halaman memerlukan cache atau tidak
- Keterbatasan hardware dan software pada sistem operasi komputer yang menjalankan Web server.

Catatan:
Ketika web server dekat dengan atau di atas batas-batasnya, ia menjadi tidak responsif.

Gejala Kelebihan Beban Kerja

Gejala-gejala dari suatu web server yang mengalami kelebihan beban kerja antara lain sebagai berikut:

- Adanya penundaan pada saat melayani permintaan antara 1 detik hingga beberapa ratus detik
- Web server selalu mengembalikan sebuah kode kesalahan HTTP, seperti 500, 502, 503, 504, 408 atau bahkan 404(untuk kondisi overload).
- Server web menolak atau mereset koneksi TCP sebelum mengirimkan konten apapun.
- Dalam kasus yang sangat langka, web server mengembalikan hanya sebagian dari konten yang diminta. Perilaku ini dapat dianggap sebagai bug, bahkan jika itu biasanya muncul sebagai gejala overload.

Penyebab Kelebihan Beban Kerja

Ada beberapa faktor yang menyebabkan web server mengalami kelebihan beban kerja, antara lain karena:

- Terlalu banyak lalu lintas web yang sah. Ribuan atau bahkan jutaan klien melakukan koneksi kesitus web dalam interval waktu yang hampir sama;
- Adanya serangan *Distributed Denial of Service*. Sebuah serangan *denial-of-service* (*DoS attack*) merupakan upaya untuk membuat sumber daya komputer atau jaringan tidak tersedia untuk pengguna yang dimaksudkan;
- *Worm* pada komputer yang kadang-kadang menyebabkan lalu lintas pada jaringan tidak normal karena banyaknya komputer yang terinfeksi;
- *Virus XSS* yang dapat menyebabkan lalu lintas tinggi karena jutaan browser atau web server yang terinfeksi;
- *Internet Bots*. Lalu Lintas transaksi tidak disaring atau tidak dibatasi pada situs web yang besar dengan sumber daya(bandwith, dan lain-lain) yang minim.
- *Internet* (jaringan) *slowdowns*, sehingga permintaan klien dilayani lebih lambat dan jumlah sambungan meningkat sehingga mencapai batas beban server;
- Kegagalan perangkat keras atau perangkat lunak, kegagalan pada back-end (misalnya, basis data).

Teknik Mengatasi kelebihan Beban Kerja pada Web Server

Untuk mengatasi batas beban kerja di atas dan mencegah overload, maka beberapa teknik di bawah ini dapat dilakukan, seperti:

- Mengelola lalu lintas jaringan, dengan menggunakan:
  - ✓ Firewall untuk memblokir lalu lintas yang tidak diinginkan yang berasal dari sumber IP yang tidak valid dan memiliki pola yang meragukan;

- ✓ Pengaturan lalu lintas HTTP untuk membatalkan, mengarahkan atau menulis ulang permintaan yang memiliki pola HTTP yang tidak tepat;
- ✓ Manajemen bandwith, dalam rangka merapikan penggunaan jaringan;

Penggunaan web cache;

- Penggunaan berbagai nama domain untuk melayani konten yang berbeda (statis dan dinamis) pada server web yang terpisah, contoh:
  - ✓ http://images.example.com
  - ✓ http://www.example.com
- Menggunakan nama domain atau komputer yang berbeda untuk memisahkan file besar dari file berukuran kecil dan menengah; idenya adalah untuk dapat sepenuhnya men-cache file berukuran kecil dan menengah dan melayani secara efisien file yang berukuran besar (lebih dari 10 - 1000 MB) dengan menggunakan pengaturan yang berbeda;
- Menggunakan banyak web server (program) per komputer, masing-masing terikat sendiri sesuai dengan kartu jaringan dan alamat IP;
- Menggunakan banyak web server (komputer) yang dikelompokkan bersama sehingga mereka bertindak atau dipandang sebagai satu server web yang besar;
- Menambahkan sumber daya perangkat keras (yaitu RAM, disk) untuk setiap komputer;
- Menggunakan lebih efisien program komputer pada web server.

3.2 HTTP Request dan HTTP Response

3.2.1 HTTP Request

Proses permintaan layanan dari client ke server adala melalui pengiriman suatu pesan. Baris pertama dari pesan tersebut adalah penggunaan metode pada sumber, identifikasi sumber dan protokol yang digunakan.

Request = *Request-Line*
((general-header
| request-header
| entity-header) CRLF)
CRLF
[message-body]

3.2.1.1 *Request-Line* (Baris Permintaan)

Baris permintaan diawali dengan penggunaan metode (*method*), diikuti oleh *request URI* dan versi protokol, dan diakhiri dengan CRLF (*carriage return linefeed*).

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

Metode(*method*)

Metode token menunjukkan metode yang akan dilakukan pada sumber daya yang diidentifikasi oleh Request-URI. Metode ini bersifat *case sensitive.*

HTTP menetapkan metode (kadang disebut "verbs") yang menunjukkan tindakan yang ingin dilakukan terhadap **sumber** teridentifikasi. Hal yang diwakili sumber ini, berupa data yang sudah ada atau data yang diciptakan secara dinamis bergantung pada implementasi browser. Biasanya sumber ini berkaitan dengan berkas atau keluaran dari berkas pelaksana yang menetap di browser. Himpunan metode umum untuk HTTP/1.1 didefinisikan di bawah ini.

Method = *"OPTIONS"*
| *"GET"*
| *"HEAD"*
| *"POST"*
| *"PUT"*
| *"DELETE"*
| *"TRACE"*
| *"CONNECT"*
| *extension-method*

extension-method = token

a. OPTION

Metode OPTION merupakan permintaan informasi tentang pilihan komunikasi yang tersedia pada rantai request/response yang diidentifikasi oleh Request-URI. Metode ini memungkinkan klien untuk menentukan pilihan atau persyaratan yang terkait dengan sumber daya, atau kemampuan server.

b. GET

Metode GET berarti mengambil informasi (dalam bentuk suatu entitas) yang diidentifikasi oleh request-URI. Jika request-URI mengacu pada proses menghasilkan data, maka proses tersebut mengembalikan beberapa data yang dipandang sebagai sebuah entitas dalam melakukan response. Data tersebut bukan data teks dari sumber yang diproses melainkan informasi (output) dari proses tersebut.

c. HEAD

Metode HEAD identik dengan GET kecuali server TIDAK HARUS mengembalikan messages-body dalam respon. Metode ini dapat digunakan untuk memperoleh meta-informasi tentang entitas yang terkandung dalam proses

permintaan, tanpa perlu mentransfer seluruh konten. Metode ini sering digunakan untuk pengujian link hypertext untuk validitas, aksesibilitas, dan modifikasi terakhir.

d. POST

Permintaan POST digunakan untuk mengirim data ke server untuk diproses dalam beberapa cara, seperti oleh script CGI. Akan tetapi fungsi yang sebenarnya dilakukan dengan metode POST ditentukan oleh server dan biasanya bergantung pada request-URI.

e. PUT

Perbedaan mendasar antara metode POST dan PUT tercermin pada penggunaan request-URI. URI dalam permintaan POST mengidentifikasi sumber daya yang akan menangani entitas tertutup. Sebaliknya, URI dalam permintaan PUT mengidentifikasi entitas disertai permintaan (request).

f. DELETE

Metode DELETE adalah metode untuk menghapus sumber daya tertentu pada server sesuai dengan identifikasi dari Request-URI

g. TRACE

Metode TRACE digunakan untuk memanggil remote lapisan-aplikasi, *loop-back* dari pesan permintaan. Penerima akhir dari permintaan HARUS mencerminkan pesan yang diterima kembali ke klien sebagai konten dari respon 200 (OK). Metode TRACE memungkinkan klien untuk melihat apa yang sedang diterima di ujung lain dari rantai permintaan dan menggunakan data tersebut untuk pengujian atau informasi diagnosis.

h. CONNECT

Menukarkan koneksi permintaan dengan TCP/IP transparan, biasanya untuk memfasilitasi komunikasi

terenkripsi SSL (HTTPS) melalui proksi HTTP tak terenkripsi.

i. PATCH

Menerapkan modifikasi parsial terhadap sumber.

Daftar metode yang diizinkan oleh sumber daya bisa ditetapkan dalam sebuah kolom header. Setelah diterima oleh server maka server mengirimkan kode balasan apakah metode diperbolehkan saat ini pada sumber daya. Sebuah server asal harus *(SHOULD)* mengembalikan kode status 405 (metode tidak diizinkan) jika metode ini dikenal dengan server asal tetapi tidak diperbolehkan untuk sumber daya yang diminta, dan 501 *(Not Implemented)* jika metode ini tidak diakui atau tidak diimplementasikan oleh server asal. Metode GET dan HEAD harus *(MUST)* didukung oleh semua server. Semua metode di atas bersifat optional, namun jika metode di atas diterapkan maka harus disesuaikan dengan semantik yang sama.

Request-URI

Request-URI digunakan untuk mengidentifikasi sumber daya pada server atau gateway dengan mengikuti aturan atau format tertentu.

Format Request URI

Request-URI = "*" | *absoluteURI* | *abs_path* | *authority*

Tanda bintang "*" berarti bahwa permintaan tersebut tidak berlaku untuk sumber daya tertentu, tetapi untuk server itu sendiri. Sementara untuk bentuk ***absoluteURI*** dibutuhkan ketika permintaan sedang dilakukan untuk proxy. Proxy diminta untuk meneruskan permintaan atau layanan dari sebuah cache yang benar dan mengembalikan respon. Pada tahap ini, proxy mungkin meneruskan permintaan tersebut ke proxy lain atau langsung ke server yang ditentukan oleh *absoluteURI*. Untuk menghindari permintaan yang berulang,

proxy HARUS dapat mengenali semua nama server(termasuk alias), variasi lokal, dan alamat IP numerik. Contoh *request-line* adalah:

GET HTTP/1.1 http://www.w3.org/pub/WWW/TheProject.html

Untuk memungkinkan transisi ke absoluteURI disemua permintaan pada versi HTTP, maka semua HTTP/1.1 server harus menerima bentuk *absoluteURI* dalam request, meskipun HTTP/1.1 pada klien hanya akan menghasilkan *absoluteURI* dalam melakukan request untuk proxy. Bentuk **otoritas** ***(authority)*** hanya digunakan pada metode CONNECT. ***abs_path*** sebagai path absolut dari URI harus dikirimkan sebagai *request-URI* dan lokasi jaringan dari URI(otoritas) harus ditransmisikan pada bagian *Host header*. Sebagai contoh, klien yang ingin mengambil sumber daya pada sebuah server asal akan membuat koneksi TCP ke port 80 dari host www.w3.org dan mengirimkan baris permintaan:

GET / pub / WWW / TheProject.html HTTP/1.1
Host : www.w3.org

Perhatikan bahwa path absolut tidak boleh kosong, jika komponen yang disebut di atas tidak ada dalam URI maka akan dipandang sebagai "/" (root server).

3.2.1.2 Identifikasi Sumber Daya

Sumber daya yang tepat dan sesuai dengan *request* pada protokol HTTP ditentukan dengan cara memeriksa *request-URI* dan bagian pada *host header*. Sebuah server tidak membedakan sumber daya berdasarkan host pada request (kadang-kadang bisa juga merujuk pada virtual host) melainkan harus menggunakan aturan. Aturan untuk menentukan sumber daya pada HTTP/1.1 request, adalah sebagai berikut:

1. Jika request-URI adalah sebuah absoluteURI dan host merupakan bagian dari requet-URI maka nilai pada *host header* harus diabaikan

2. Jika request-URI bukan absolute URI dan request memuat *host header* maka host ditentukan oleh bagian *host header*
3. Jika host yang ditentukan pada aturan 1 dan 2 bukan host yang valid dalam server maka pesan kesalahan pada *request response* adalah 400 (*Bad Request*).

Catatan:
Pada penerimaan request di HTTP/1.0 tidak memiliki field host pada header.

3.2.1.3 Header Permintaan (*Request Header Fields*)

Request header fields mengizinkan klien untuk menyampaikan informasi tambahan tentang permintaan dan klien itu sendiri kepada server. Isi dari field ini dapat bertindak sebagai pengubah permintaan. Fitur yang terdapat pada *request-header* adalah:

request-header = *Accept*
| Accept-Charset
| Accept-Encoding
| Accept-Language
| Authorization
| Expect
| From
| Host
| If-Match
| If-Modified-Since
| If-None-Match
| If-Range
| If-Unmodified-Since
| Max-Forwards
| Proxy-Authorization

| Range
| Referer
| TE
| User-Agent

Catatan:
Penggunaan dari request-header ini dapat dilihat pada ***Browser Request.***

3.2.2 HTTP *Response*

Setelah menerima dan menginterpretasikan sebuah permintaan layanan dari *client* maka *server* membalas (merespon) dengan pesan HTTP balasan.

Response = Status-Line
**((general-header*
| response-header
| entity-header) CRLF)
CRLF
[message-body]

Status - Line

Baris pertama dari pesan balasan dari server adalah status-line, penyesuaian dari versi protocol dan diikuti oleh numeric kode status.

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

Kode Status

Kode status terdiri dari 3 digit, dimana digit pertama dari kode status adalah definisi dari *class response* sementara dua digit terakhir adalah aturan yang mengikut pada digit pertama. Kode status terbagi dalam lima *class response* yaitu *informational* (1xx), sukses (2xx), *redirection* (3xx), *client error* (4xx) dan *server error* (5xx).

Status-Code = "100" : Continue
| *"101" : Switching Protocols*
| *"200" : OK*
| *"201" : Created*
| *"202" : Accepted*
| *"203" : Non-Authoritative Information*
| *"204" : No Content*
| *"205" : Reset Content*
| *"206" : Partial Content*
| *"300" : Multiple Choices*
| *"301" : Moved Permanently*
| *"302" : Found*
| *"303" : See Other*
| *"304" : Not Modified*
| *"305" : Use Proxy*
| *"307" : Temporary Redirect*
| *"400" : Bad Request*
| *"401" : Unauthorized*
| *"402" : Payment Required*
| *"403" : Forbidden*
| *"404" : Not Found*
| *"405" : Method Not Allowed*
| *"406" : Not Acceptable*
| *"407" : Proxy Authentication Required*
| *"408" : Request Time-out*
| *"409" : Conflict*
| *"410" : Gone*
| *"411" : Length Required*

| *"412" : Precondition Failed*
| *"413" : Request Entity Too Large*
| *"414" : Request-URI Too Large*
| *"415" : Unsupported Media Type*
| *"416" : Requested range not satisfiable*
| *"417" : Expectation Failed*
| *"500" : Internal Server Error*
| *"501" : Not Implemented*
| *"502" : Bad Gateway*
| *"503" : Service Unavailable*
| *"504" : Gateway Time-out*
| *"505" : HTTP Version not supported*
| *extension-code*

extension-code = 3DIGIT

*Reason-Phrase = *<TEXT, excluding CR, LF>*

Catatan:

- *Aplikasi HTTP tidak harus mengerti arti dari semua daftar kode status meskipun pemahaman seperti ini jelas dibutuhkan.*
- *Aplikasi harus memahami kelas dari setiap kode status, seperti yang ditunjukkan oleh angka pertama dan memperlakukan setiap respon yang belum diakui (unrecognized) setara dengan status x00 dari kelas itu dan pengecualian bahwa respon yang belum diakui sebagai cache.*

Header Response (Response Header Fields)

Response header fields memperbolehkan server untuk menyampaikan informasi tambahan tentang respon yang tidak dapat ditempatkan pada *Status Line.* Informasi yang dimaksud dapat berupa informasi tentang server dan tentang akses lebih lanjut ke sumber yang diidentifikasi oleh request-URI.

response-header = Accept-Ranges
| *Age*
| *ETag*
| *Location*
| *Proxy-Authenticate*
| *Retry-After*
| *Server*
| *Vary*
| *WWW-Authenticate*

Catatan:
Untuk Bacaan dan telaah lebih lanjut mengenai HTTP Request dan Response dapat dilihat pada (http://www.w3.org/Protocols/rfc2616).

3.3 Penanganan HTTP Request dan HTTP Response

Untuk menggambarkan proses penanganan request dari klien dapat dilihat pada siklus berikut ini.

1. Pengguna memasukkan alamat pada browser.

   Misalnya URL yang diketikkan oleh pengguna pada browser adalah:

   http://www.uin-alauddin.ac.id/manafile.html

2. Browser membuat *HTTP request* dan mengirimkannya ke server.

   Pada saat melakukan koneksi dengan www.uin-alauddin.ac.id, URL di atas akan diterjemahkan oleh browser kedalam bentuk permintaan sebagai berikut:

```
GET /path/manafile.html HTTP/1.1
Host: uin-alauddin.acid
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
```

3. **Web Server memparsing *HTTP request***

   Web server pada www.uin-alauddin.ac.id akan menambahkan path untuk akses ke root direktori. Pada *server Apache,* path direktori biasanya */home/www* sementara pada UNIX biasanya */var / www.* Maka hasil akhirnya adalah definisi path dari file lokal yaitu:

   /home/www/path/manafile.html

   Proses parsing selanjutnya adalah menentukan tindakan apa yang harus dilakukan oleh server. Terdapat tiga informasi penting yang perlu diketahui dari *HTTP request* yaitu method, jenis dokumen dan protocol HTTP yang digunakan.

4. Web server membaca informasi lain pada jaringan (bila perlu)

   Informasi yang dimaksud mengenai User-agent, Accept, Accept-Language, Accept-Encoding, Accept-Charset dan Connection.

5. Web Server melayani permintaan(dengan asumsi tidak ada kesalahan pada HTTP request), kemudian mengolah dan mengirim balasan menggunakan *HTTP response.*

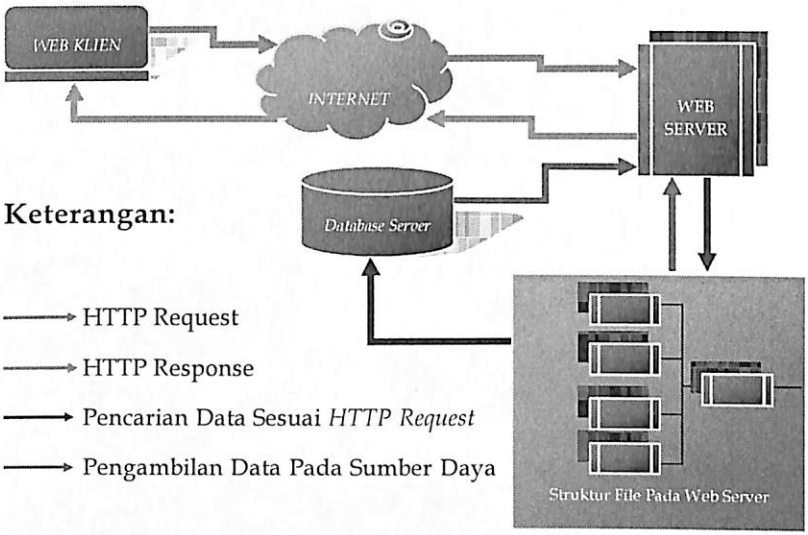
Bentuk dari HTTP response(bila filenya ditemukan)

HTTP/1.1 200 OK
Date: Sat, 18 Mar 2012 20:35:35 GMT
Cache-Control : private, max-age=0
Content-Type: text/html
Server: gws
Content-Length: 141
X-Cache : MISS from
Via : 1.0 .:8001 (squid)
Connection : keep-alive
<html>
<head>
<title>contoh http response</title>
</head>
<body>
<h1>contoh dokumen</h1>
</body>
</html>

Server menghasilkan respon yang mencakup versi protokol HTTP, kode status HTTP, dan frase alasan yang dipisahkan oleh spasi. Hal ini biasanya diikuti oleh sejumlah header dan akhir dari header ditunjukkan oleh baris kosong. Pada HTTP response terdapat beberapa elemen penting yang perlu diperhatikan, antara lain cache header, server header, via header dan content type.

6. Broswser menerima *HTTP response*, memprosesnya dan hasil akhirnya ditampilkan pada layar monitor pengguna.

Proses di atas dapat di gambarkan sebagai berikut:



Gambar 3.1
Proses Penanganan HTTP Request dan HTTP Response

3.4 Arsitektur Web Server

Web server sebagai bagian dari teknologi World Wide Web memiliki arsitektur yang beragam, hal ini bergantung pada teknologi server yang digunakan. Untuk mendalami pengetahuan tentang pengembangan arsitektur web server, maka ada beberapa poin penting yang perlu untuk diketahui yaitu tujuan desain arsitektur, prinsip arsitektur web, fitur tambahan yang bisa dikembangkan dan model arsitektur.

3.4.1 Prinsip Arsitektur Web Server

Sejumlah prinsip arsitektur umum berlaku untuk semua arsitektur web server. Terdapat beberapa prinsip utama dalam arsitektur web server.

1. Spesifikasi Ortogonal

   Identifikasi, interaksi dan representasi adalah konsep orthogonal yang berarti bahwa teknologi yang digunakan

untuk identifikasi, interaksi dan representasi dapat berkembang secara mandiri.

Misalnya : Indentifikasi Sumber daya dengan URI.

URI dapat dipublikasikan tanpa membangun pernyataan apapun dari sumber daya. Sebuah sintaks URI memungkinkan agen pengguna berfungsi dalam banyak kasus tanpa mengetahui spesifik dari skema URI. Dalam banyak kasus seseorang dapat mengubah representasi dari sumber daya tanpa mengganggu referensi ke sumber daya lain (misalnya, dengan menggunakan negosiasi konten).

Ketika spesifikasi dua ortogonal, seseorang dapat mengubah satu tanpa memerlukan perubahan ke yang lain, bahkan jika salah satu memiliki ketergantungan pada yang lain. Sebagai contoh, meskipun spesifikasi HTTP tergantung pada spesifikasi URI, namun dua ortogonal dapat berkembang secara mandiri. Proses ortogonal ini meningkatkan fleksibilitas dan ketahanan dari Web. Sebagai contoh, seseorang mungkin merujuk oleh URI untuk gambar tanpa mengetahui tentang format yang dipilih untuk mewakili gambar. Hal ini telah memfasilitasi pengenalan format gambar seperti PNG dan SVG tanpa mengganggu referensi yang ada untuk sumber gambar.

2. Ekstensibility

Informasi dalam Web dan teknologi yang digunakan untuk mewakili perubahan informasi dari waktu ke waktu. Diperpanjang adalah milik teknologi yang mempromosikan evolusi tanpa mengorbankan interoperabilitas. Beberapa contoh teknologi yang sukses dirancang untuk memungkinkan perubahan sementara meminimalkan gangguan meliputi:

- spesifikasi skema URI yang telah ditentukan;
- penggunaan seperangkat jenis media internet terbuka di mail dan HTTP untuk menentukan interpretasi dokumen;
- pemisahan tata bahasa XML dan set terbuka dari namespaces XML untuk nama elemen dan atribut;
- model dalam Cascading Style Sheets (CSS), XSLT 1.0, dan SOAP;
- plug in pada user agent plug-in.

3. Penanganan Kesalahan

   Kesalahan terjadi pada sistem informasi jaringan. Kondisi kesalahan dapat ditandai (misalnya, well-formedness kesalahan dalam XML atau kesalahan klien 4xx di HTTP) atau muncul tak terduga. Koreksi kesalahan berarti bahwa perbaikan kondisi dalam sistem itu, seolah-olah tidak pernah terjadi kesalahan. Salah satu contoh koreksi kesalahan data yang melibatkan transmisi dalam menanggapi kegagalan jaringan sementara. Pemulihan kesalahan berarti bahwa agen tidak memperbaiki kondisi kesalahan tapi terus pengolahan dengan mengatasi fakta bahwa kesalahan telah terjadi.

4. Berbasis Interoperabilitas

   Web mengikuti tradisi penting internet dalam mendefenisikan interface penting dalam hal protokol, sintaks, semantik dan kendala urutan pesan yang dipertukarkan. Protokol dirancang untuk menjadi tangguh dalam menghadapi lingkungan yang sangat beragam dan telah membantu dan memfasilitasi aplikasi berbasis berbasis web untuk berkomunikasi melintasi perbedaan platform.

3.4.2 File Log dan Keamanan

Logging

Sebuah file log pada web server dapat menjelaskan beberapa hal penting yang berkaitan dengan akvititas pada web server. Sejumlah data yang terdapat dalam file log adalah sebagai berikut:

- Jumlah kunjungan dan jumlah pengunjung unik
- Lama kunjungan dan kunjungan terakhir
- Hari, jam dan minggu
- Domain/negara pengunjung
- Daftar host
- Jumlah halaman yang dikunjungi
- Halaman yang terbanyak dilihat, masuk dan keluar
- Jenis file
- Sistem operasi yang digunakan
- Browser yang digunakan
- Robot
- HTTP referrer
- Search engine, frase dan kata kunci yang digunakan untuk menemukan situs web
- HTTP error

Keamanan

Berbagai serangan hacking telah membuktikan bahwa keamanan tetap menjadi isu yang paling penting untuk setiap bisnis yang melakukan operasinya secara online. *Web server* adalah salah satu wajah publik yang paling ditargetkan dari sebuah organisasi, karena data sensitif mereka terkadang disimpan pada *web server*. Mengamankan web server web adalah sama pentingnya dengan mengamankan website atau aplikasi web itu sendiri dan jaringan di sekitarnya. Apabila

sebuah perusahaan memiliki aplikasi web yang aman akan tetapi tidak memiliki *web server* yang aman atau sebaliknya, maka dengan sendirinya mereka menempatkan bisnis pada risiko yang sangat besar. Meskipun mengamankan *web server* dapat menjadi operasi menakutkan dan memerlukan keahlian khusus, ini bukan tugas yang mustahil. Untuk menangani masalah keamanan *web server* maka dapat diambil beberapa langkah yang diperlukan.

1. Menghapus Layanan yang Tidak Perlu

Standar instalasi sistem operasi dan konfigurasi yang tidak aman. Dalam instalasi default yang khas, banyak layanan jaringan yang tidak akan digunakan dalam konfigurasi web server yang diinstal, seperti layanan remote registry, layanan print server, dan lain-lain. Semakin banyak service (layanan) yang berjalan pada sistem operasi, port akan lebih dibiarkan terbuka, sehingga meninggalkan pintu lebih terbuka bagi pengguna yang jahat terhadap penyalahgunaan. Matikan atau nonaktifkan semua layanan yang tidak perlu, jadi ketika suatu saat server reboot, layanan tersebut tidak dimulai secara otomatis. Mematikan layanan yang tidak perlu juga akan memberikan dorongan ekstra untuk akselerasi pada server dan membebaskan beberapa sumber daya perangkat keras.

2. Remote Akses

Meskipun saat ini tidak praktis, akan tetapi ada baiknya kalau administrator server harus login ke web server lokal. Jika remote akses diperlukan, maka harus dipastikan bahwa sambungan remote dijamin dengan benar, dengan menggunakan protokol tunneling dan enkripsi. Menggunakan token keamanan dan single sign lainnya pada peralatan dan perangkat lunak, adalah praktek keamanan yang sangat baik. Remote akses juga harus dibatasi pada jumlah IP dan akun tertentu. Hal ini juga sangat penting untuk tidak meng-

gunakan komputer publik atau jaringan publik untuk mengakses server jarak jauh, seperti di kafe internet atau jaringan nirkabel publik.

3. Memisahkan Pengembangan/Pengujian/Produksi

Sebuah aplikasi web, akan lebih mudah dan lebih cepat bagi pengembang untuk mengembangkan versi terbaru dari aplikasi web pada server produksi, sangat umum bahwa pembangunan dan pengujian aplikasi web dilakukan langsung pada server produksi itu sendiri. Ini adalah kejadian umum di internet untuk menemukan versi yang lebih baru dari situs web tertentu, atau beberapa konten yang tidak harus tersedia untuk umum di direktori seperti/ test / , / new / atau sub direktori yang serupa, hal ini disebabkan oleh karena aplikasi web seperti berada dalam tahap awal pembangunan sehinga mereka cenderung memiliki sejumlah kerentanan, kurangnya validasi input dan tidak menangani pengecualian tepat. Aplikasi ini dapat dengan mudah ditemukan dan dieksploitasi oleh pengguna berbahaya, dengan menggunakan alat yang tersedia gratis di internet.

Untuk lebih memudahkan pengembangan dan pengujian aplikasi web, pengembang cenderung untuk mengembangkan aplikasi internal tertentu yang memberi mereka akses istimewa ke aplikasi web, database dan sumber daya web server lainnya, dimana pengguna umum tidak mendapatkan akses untuk hal tersebut. Aplikasi yang seperti ini biasanya tidak memiliki jenis pembatasan, karena mereka hanya menguji aplikasi yang diakses yang harus diakses dari pengembang saja. Sayangnya, jika pengembangan dan pengujian dilakukan pada server produksi, aplikasi tersebut dengan mudah dapat ditemukan dari pengguna berbahaya, yang bisa membantunya untuk mendapatkan akses pada server produksi.

Idealnya, pengembangan dan pengujian aplikasi web harus selalu dilakukan pada server terisolasi dari internet, dan tidak boleh menggunakan atau terhubung ke data yang real untuk aplikasi terutama database.

4. Web Konten dan Server Side Scripting

Aplikasi web atau file website dan script harus selalu berada pada partisi terpisah atau drive selain dari sistem operasi, log dan file sistem lainnya. Banyak fakta mengatakan, jika hacker yang memperoleh akses ke direktori root web, maka mereka mampu mengeksploitasi kerentanan lainnya, dan mampu melangkah lebih jauh dan meningkatkan hak-hak mereka untuk mendapatkan akses ke data pada seluruh disk, termasuk sistem operasi dan file sistem lainnya. Dari sana dan seterusnya, para pengguna yang jahat memiliki akses untuk melaksanakan setiap perintah sistem operasi, sehingga kontrol penuh dari web server berada pada tangan mereka.

5. Perizinan dan Hak Istimewa (*Permissions and Privileges*)

File dan *permission* pada layanan jaringan memainkan peran penting dalam keamanan web server. Jika mesin web server web dikompromikan melalui software layanan jaringan, pengguna berbahaya dapat menggunakan account di mana layanan jaringan berjalan untuk melaksanakan tugas, seperti mengeksekusi file tertentu. Oleh karena itu sangat penting untuk selalu memberikan hak istimewa setidaknya diperlukan untuk layanan jaringan tertentu untuk menjalankan, seperti perangkat lunak pada web server. Hal ini juga sangat penting untuk menetapkan hak minimum untuk pengguna umum dalam mengakses situs, file aplikasi web, data serta sistem database.

6. Menginstal Patch Keamanan

Meskipun memiliki perangkat lunak yang telah di*pacth*, akan tetapi itu bukan berarti web server sepenuhnya berada

dalam keadaan aman, masih sangat penting untuk memperbarui sistem operasi dan perangkat lunak lain yang berjalan di atasnya dengan patch keamanan terbaru. Insiden hacking masih terjadi karena hacker mengambil keuntungan dari eksploitasi dari server dan perangkat lunak yang belum di*patch*.

7. Memantau dan Mengaudit Server

Semua log yang ada dalam web server, idealnya harus disimpan dalam area terpisah. Semua log layanan jaringan, log akses situs, log database server (misalnya Microsoft SQL Server, MySQL, Oracle) dan log sistem operasi harus sering dipantau dan diperiksa. File log cenderung untuk memberikan semua informasi mengenai upaya serangan, dan bahkan dari serangan yang sukses, tapi sebagian besar kali ini diabaikan. Jika seseorang memperhatikan aktivitas aneh dari log, ini harus segera meningkat sehingga masalah ini dapat diselidiki untuk melihat apa yang terjadi.

8. Akun Pengguna

Akun pengguna standar yang tidak terpakai (yang diciptakan selama instalasi sistem operasi) harus dinonaktifkan. Ada juga daftar panjang software yang ketika diinstal, user yang dibuat pada sistem operasi. Akun tersebut juga harus diperiksa dengan benar dan izin perlu diubah jika diperlukan. Yang dibangun pada akun administrator harus diganti dan tidak boleh digunakan, yang sama untuk pengguna root pada saat instalasi linux/unix. Setiap administrator mengakses web server harus memiliki akun pengguna sendiri, dengan hak istimewa yang benar. Ini juga merupakan praktik keamanan yang baik untuk tidak berbagi account pengguna satu sama lain.

9. Hapus Semua Modul yang Tidak Terpakai

Sebuah instalasi default Apache memiliki sejumlah modul yang telah ditentukan untuk diaktifkan, yang dalam skenario

web server yang khas tidak digunakan, kecuali mereka secara khusus dibutuhkan. Matikan modul tersebut untuk mencegah serangan yang ditargetkan terhadap modul tersebut.

Hal yang sama berlaku untuk web server Microsoft, Internet Information Services. Secara default, IIS dikonfigurasi untuk melayani sejumlah besar jenis aplikasi, misalnya ASP, ASP.NET dan banyak lagi. Setiap ekstensi aplikasi juga harus dibatasi untuk menggunakan verba HTTP tertentu saja, mana mungkin.

10. Gunakan Alat Keamanan yang Disediakan dengan Perangkat Lunak Web Server

Microsoft merilis sejumlah alat untuk membantu administrator melakukan instalasi aman IIS web server seperti scan URL. Ada juga modul yang disebut mod_security untuk Apache. Meskipun mengkonfigurasi alat tersebut merupakan proses yang membosankan dan dapat memakan waktu, terutama dengan aplikasi web kustom, mereka menambahkan sedikit tambahan keamanan.

11. Mendapatkan Informasi Terbaru

Saat ini, informasi dan tips tentang sistem operasi dan perangkat lunak yang digunakan dapat ditemukan secara bebas di internet. Hal ini sangat penting untuk tetap mendapatkan informasi dan belajar tentang serangan baru dan alat yang digunakan. Dengan membaca majalah yang terkait keamanan dan berlangganan newsletter, forum atau jenis layanan masyarakat lainnya.

12. Gunakan Scanner

Scanner adalah alat praktis yang membantu mengotomatisasi dan mempermudah proses mengamankan web server dan aplikasi web. Acunetix Web Vulnerability Scanner misalnya merupakan aplikasi yang dilengkapi dengan port scanner, yang bila diaktifkan akan melakukan scan

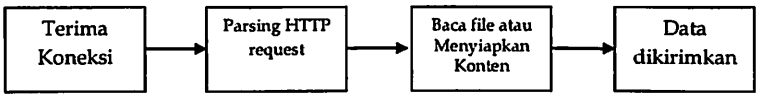
terhadap port yang ada pada we server. Mirip dengan scan pada keamanan jaringan, Acunetix WVS juga meluncurkan sejumlah pemeriksaan keamanan yang canggih terhadap port yang terbuka dan layanan jaringan yang berjalan pada web server web.

Acunetix Web Vulnerability scanner menjamin situs web dan keamanan web server dengan memeriksa SQL Injection, Cross Site Scripting, masalah konfigurasi web server dan kerentanan lainnya. Ia memeriksa kekuatan password pada halaman otentikasi dan audit secara otomatis bentuk, konten Web 2.0 yang dinamis dan aplikasi web lainnya.

3.4.3 Model Arsitektur Web Server

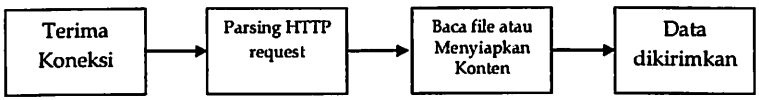
- **Multi-Proses (Apache pada Unix)**

Proses 1



. . .

Proses N

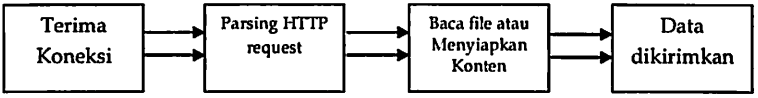


Gambar 3.2 *Model Arsitektur Multi Proses*

Ciri Utama pada model ini, adalah:

- Memanfaatkan beberapa prosesor
- Mudah untuk debug
- Komunikasi antar proses agak sulit dan mahal
- Penggunaan memori yang berbiaya tinggi

- **Multi Thread (Apache pada NT/XP)**

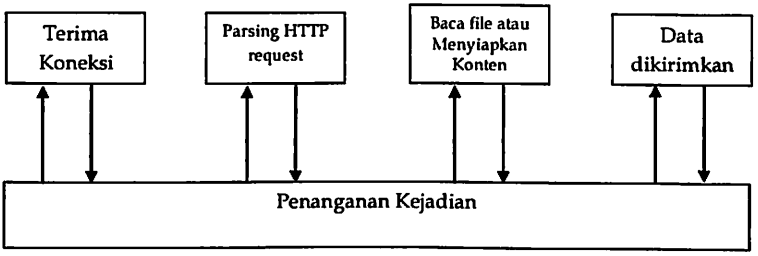


Gambar 3.3 *Model Arsitektur Multi Thread*

Ciri Utama pada model ini, adalah:

- Memanfaatkan beberapa thread untuk kinerja yang lebih baik
- Mudah untuk mengubah kebijakan threading
- Harus melakukan sinkronisasi, untuk menghindari ras data
- Pemanfaatan sumber daya (kernel dan user-level)
- pemakaian memori, switch konteks dan startup
- Blocking pada I / O yang dapat menyebabkan *deadlock*

- **Single process Event-Driven**

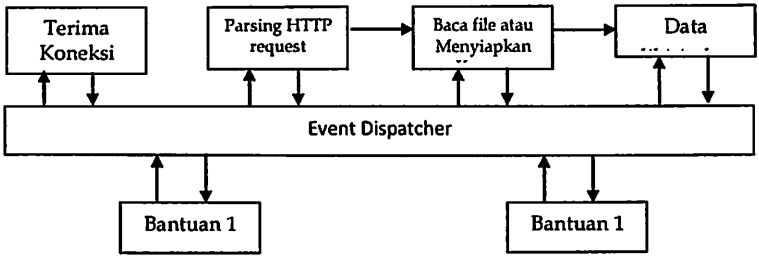


Gambar 3.3 *Model Arsitektur Single Process Event Drivent*

Ciri Utama pada model ini, adalah :

- Menggunakan pemilihan untuk memeriksa file yang siap dideskripsikan
- Mesin menentukan bagaimana untuk pindah ke tahap proses selanjutnya
- Tidak ada konteks switching, sinkronisasi dan ruang alamat tunggal
- Sistem operasi modern tidak memberikan dukungan yang cukup untuk operasi *disk asynchronous*

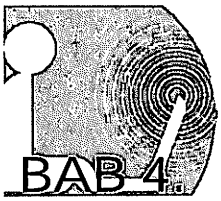
- **Terima Koneksi**



Gambar 3.4

Model Arsitektur Asymetric Multi-Proses Event-Driven

Serupa dengan Proses Single Event-Driven tetapi dengan bantuan khusus untuk memblokir I / O (misalnya, permintaan disk).{"}



Desain WEB

Pada bagian akhir buku ini akan dipaparkan tentang konsep dasar dan pengetahuan yang berkaitan dengan desain web serta perbandingan beberapa teknologi yang akan digunakan.

- Interaksi Manusia dan Komputer
- Bahasa *Markup* (HTML, XHTML dan XML)
- Pemilihan Database (Mysql dan PostgreSQL)
- Penggunaan Bahasa Pemrograman (*Client Side Scripting dan Server Side Scripting*)
- Rancangan Tampilan
- Gaya Lebar Bahasa (CSS dan XSL)
- Teknologi Multimedia (*Flash* dan *Silverlight*)

4.1 Interaksi Manusia dan Komputer

Interaksi manusia dan komputer (bahasa Inggris : *Human Computer Interaction/HCI)* melibatkan studi, perencanaan, dan desain interaksi antara manusia dan komputer itu sendiri. Hal ini sering dianggap sebagai perpaduan antara ilmu komputer, ilmu perilaku, ilmu desain dan beberapa bidang studi lainnya. Istilah ini diciptakan oleh Card Moran dan Newell dalam buku mereka, *"The Psychology of Human Computer Interaction". The Association for Computing Machinery* mendefinisikan interaksi manusia-komputer sebagai "disiplin berkaitan dengan evaluasi, desain dan implementasi

sistem komputer interaktif untuk digunakan manusia dan dengan studi fenomena besar di sekitar mereka." Tujuan akhir dari HCI adalah untuk menciptakan kepuasan pengguna walaupun pada dasarnya ada perbedaan pada tingkat kepuasan pada tiap-tiap pengguna.

Interaksi manusia dan komputer (IMK) sebagai studi yang mempelajari hubungan antara manusia dan mesin, maka perkembangan ilmu ini sangat bergantung pada pemahaman yang mendasar akan ilmu mesin dan pengetahuan akan sisi manusia. Di sisi mesin, teknik dalam komputer grafik, sistem operasi, bahasa pemrograman dan lingkungan pengembangan yang relevan merupakan disiplin pengetahuan utama yang perlu difahami. Sedangkan pada sisi manusia, teori komunikasi, ilmu linguistik, ilmu sosial, psikologi kognitif serta pemahaman tentang faktor-faktor yang mempengaruhi kepuasan pengguna komputer yang relevan. Karena sifat multidisiplin dari IMK, maka tidak menutup kemungkinan bagi orang dengan latar belakang yang berbeda dapat memberikan kontribusi untuk keberhasilannya. IMK juga kadang-kadang disebut sebagai Interaksi Manusia-Mesin (*Man-Machine Interaction*).

Perhatian kepada interaksi manusia-mesin ini sangat penting, karena kesalahan pada rancangan antarmuka manusia dan mesin dapat menyebabkan masalah yang tak terduga. Sebuah contoh klasik dari hal ini adalah kecelakaan pada Three Mile Island (1979) dimana hasil akhir dari investigasi menyimpulkan bahwa desain antarmuka manusia-mesin itu setidaknya sebagian besar bertanggung jawab atas terjadinya bencana pada industri nuklir tersebut. Contoh kongkrit dari adanya hubungan antara manusia dan komputer adalah pada saat karakter atau objek ditampilkan oleh perangkat lunak pada layar monitor komputer pribadi, masukan yang diinput oleh pengguna melalui perangkat keras seperti keyboard dan mouse dan interaksi pengguna

lain dengan skala besar sistem komputerisasi seperti pada sistem informasi penerbangan. Interaksi ini terjadi melalui sebuah antarmuka (*interface*), yang meliputi perangkat keras dan perangkat lunak. Dalam interaksi manusia dengan komputer terdapat beberapa panca indera digunakan untuk dapat berinteraksi. Manusia mewujudkan fisiologi yang diperlukan untuk menyerap informasi dalam bentuk suara. Sama seperti mata dapat melihat berbagai variasi cahaya, brightness dan kontras--telinga sebagai salah satu panca indera mampu membedakan perubahan suara melalui perubahan tibre, kenyaringan, dan pitch. Pikiran kemudian dapat mengasosiasikan suara ini dengan peristiwa, objek, atau gagasan abstrak.

4.1.1 Tujuan Interaksi Manusia dengan Komputer

IMK bertujuan untuk memudahkan manusia dalam mengoperasikan komputer dan mendapatkan berbagai umpan balik yang ia perlukan selama ia bekerja pada sebuah sistem komputer. Para perancang antarmuka manusia dan komputer berharap agar sistem komputer yang dirancangnya dapat bersifat akrab dan ramah dengan penggunanya (user friendly). Tujuan jangka panjang dari IMK adalah untuk merancang sistem yang meminimalkan penghalang antara model kognitif manusia dari apa yang mereka ingin capai dan pemahaman komputer dari tugas pengguna. Praktisi profesional di IMK biasanya desainer peduli dengan aplikasi praktis dari metodologi desain untuk masalah di dunia nyata. Pekerjaan mereka sering berkisar sekitar merancang grafis antarmuka pengguna dan antarmuka website. Para peneliti di IMK tertarik untuk mengembangkan metodologi desain baru, bereksperimen dengan perangkat keras baru, prototype baru untuk sistem perangkat lunak, menjelajahi paradigma baru untuk interaksi, dan model pengembangan serta teori interaksi.

IMK berbeda dari faktor manusia(ergonomi) karena IMK lebih berfokus pada cara pengguna bekerja dengan komputer, bukan mesin-mesin lainnya atau artefak dirancang. Ada juga fokus dalam IMK tentang bagaimana menerapkan perangkat lunak komputer dan mekanisme perangkat keras untuk mendukung interaksi manusia-komputer. Jadi, faktor manusia adalah istilah yang lebih luas karena IMK dapat digambarkan sebagai faktor manusia komputer, meskipun beberapa ahli mencoba untuk membedakan antara manusia dan komputer itu sendiri. Tiga bidang studi memiliki tumpang tindih substansial dengan IMK bahkan sebagai fokus pergeseran penyelidikan. Dalam studi manajemen informasi pribadi (Privacy Information Management), interaksi manusia dengan komputer ditempatkan dalam konteks informasi yang lebih besar dan orang mungkin bekerja dengan berbagai bentuk informasi, beberapa berbasis komputer, banyak juga yang tidak (misalnya, papan tulis, notebook, catatan tempel) untuk memahami dan mempengaruhi perubahan yang diinginkan dalam dunia mereka. Dalam Computer Supported Cooperative Work (CSCW), penekanan ditempatkan pada penggunaan sistem komputasi untuk mendukung kerja kolaboratif dari sekelompok orang. Prinsip-prinsip Human Interaction Management(HIM) memperluas cakupan CSCW ke tingkat organisasi dan dapat dilaksanakan tanpa penggunaan sistem komputer.

4.1.2 Faktor yang Mempengaruhi IMK

Cara manusia berinteraksi dengan komputer berkembang semakin pesat. Interaksi manusia-komputer dipengaruhi oleh sifat komputasi masa depan yang meliputi:

- Penurunan biaya perangkat keras yang mengarah ke memori yang lebih besar dan sistem lebih cepat;
- Miniaturisasi perangkat keras yang mengarah ke portabilitas;

- Pengurangan kebutuhan daya yang mengarah ke portabilitas;
- Teknologi tampilan baru yang mengarah pada kemasan perangkat komputasi dalam bentuk-bentuk yang baru;
- Perangkat keras khusus yang mengarah ke fungsi baru;
- Peningkatan pengembangan jaringan komunikasi dan komputasi terdistribusi;
- Semakin luasnya penggunaan komputer, terutama oleh orang yang di luar profesi komputasi ;
- Makin meningkatnya inovasi dalam teknik input (misalnya, suara, gerakan , pena), dikombinasikan dengan menurunkan biaya

Berdasarkan sifat dari komputasi masa depan, maka penelitian untuk IMK diharapkan mencakup karakteristik sebagai berikut :

- **Ubiquitous komunikasi:** Komputer diharapkan menjadi media komunikasi baik melalui jaringan kecepatan tinggi local (LAN) maupun dalam jaringan internet (WAN). Data dan layanan komputasi yang bersifat portable akan mudah diakses dari sebagian besar lokasi pengguna.
- **Sistem yang memiliki fungsionalitas tinggi** : Sistem dapat memiliki banyak fungsi yang terkait dengan mereka. Ada begitu banyak sistem yang kebanyakan pengguna, teknis atau non-teknis, tidak punya waktu untuk mempelajarinya dengan cara tradisional (misalnya, melalui buku manual yang tebal).
- **Ketersediaan komputer grafis:** kemampuan komputer grafis seperti pengolahan citra, transformasi grafik, rendering, dan animasi interaktif

menjadi semakin luas dengan harga chip yang murah sehingga memungkinkan untuk dimasukkan dalam workstation umum dan perangkat mobile.

- **Bandwith yang lebar untuk interaksi.** Tingkat di mana manusia dan mesin berinteraksi diperkirakan akan meningkat secara substansial karena perubahan dalam kecepatan, komputer grafis, media baru, dan baru perangkat input/output. Hal ini dapat menyebabkan beberapa antarmuka yang berbeda secara kualitatif, seperti virtual reality atau video komputasi.
- **Display (Layar) yang Besar dan Tipis:** memungkinkan display yang sangat besar dan tipis, ringan, dan konsumsi daya rendah.
- **Utilitas Informasi,** utilitas informasi publik (seperti perbankan dan belanja) serta layanan industri khusus (misalnya, cuaca untuk pilot) diharapkan untuk berkembang biak.

4.1.3 Media Antarmuka Manusia dan Komputer

Media antarmuka manusia dan komputer dapat terbagi atas 2 macam, yaitu:

a. **Media Tekstual**

Adalah bentuk sederhana dialog atau komunikasi antara manusia dan komputer yang hanya berisi teks dan "kurang menarik". Salah satu contoh antarmuka manusia dan komputer berbentuk teks yang menggunakan bahasa pemrograman PASCAL adalah adalah readln dan writeln.

b. **Media GUI** (*Graphical User Interface*)

Adalah bentuk dialog atau komunikasi antara manusia dan komputer yang berbentuk grafis dan sangat atraktif. Contoh antarmuka manusia dan komputer yang berbentuk grafis menggunakan pemrograman visual.

Aturan-Aturan dalam Desain Antarmuka, yang berkaitan dengan IMK:

a. Upayakan untuk konsistensi.
   - o Urutan tindakan yang konsisten harus diminta dalam situasi yang mirip.
   - o Terminologi Identik harus digunakan pada prompt, menu, dan membantu layar.
   - o Warna yang konsisten, tata letak, kapitalisasi, font, dan sebagainya harus digunakan seluruhnya.

b. Memungkinkan pengguna untuk menggunakan jalan pintas

   Untuk meningkatkan laju singkatan menggunakan interaksi, tombol khusus, perintah tersembunyi, dan makro.

c. Penawaran informatif umpan balik

   Untuk setiap tindakan pengguna, sistem harus merespon dalam beberapa cara (dalam desain web, hal ini dapat dicapai dengan DHTML - misalnya, tombol akan membuat suara klik atau mengubah warna saat diklik untuk menampilkan sesuatu yang sebelumnya telah dieksekusi oleh pengguna).

d. Desain dialog pada saat penutupan aplikasi

   Urutan tindakan harus diatur ke dalam kelompok dengan awal, tengah, dan akhir. Umpan balik yang informatif pada penyelesaian dari sebuah tindakan pengguna akan menunjukkan aktivitas mereka telah selesai dengan sukses.

e. Penawaran pencegahan kesalahan dan penanganan kesalahan sederhana.
   - o Desain bentuk sehingga pengguna tidak dapat membuat kesalahan serius, misalnya, lebih memilih pilihan menu untuk membentuk mengisi dan tidak mengizinkan karakter abjad di bidang form inputan yang bersifat numerik.

- o Jika pengguna melakukan kesalahan, instruksi harus ditulis untuk mendeteksi kesalahan dan menawarkan instruksi sederhana, konstruktif, dan khusus untuk pemulihan.

f. Izin tindakan pemulihan.

g. Dukungan internal lokus kontrol.

h. Kurangi beban memori jangka pendek.

Sebuah studi menunjukkan bahwa manusia terkenal dapat menyimpan hanya 7 buah (plus atau minus 2) informasi dalam memori jangka pendek mereka. Kita dapat mengurangi beban memori jangka pendek dengan merancang layar di mana pilihan yang jelas terlihat, atau menggunakan pull-down menu dan ikon.

4.2 Bahasa Markup (*Markup Language*)

4.2.1 HTML dan XHTML

HTML

Detail dan aturan penulisan dokumen HTML, dapat dibaca kembali pada sub bab **1.5.**

XHTML

XHTML(*Extensible Hypertext Markup Language*) adalah bahasa markup yang awalnya diharapkan untuk menghasilkan bahasa markup yang multi platform. Untuk sebagian besar dokumen XHTML 1.0 dan dokumen HTML 4.01 hanya berbeda dalam aturan leksikal dan sintaksis. Untuk pengembangan jenis dokumen XHTML maka W3C mengeluarkan beberapa versi yang spesifik antara lain XHTML 1.0, XHTML Basic, XHTML Modularization, dan XHTML Print.

4.2.1.1 Versi XHTML

XHTML 1.0

XHTML 1.0 adalah Rekomendasi W3C pertama untuk XHTML, sebagai lanjutan dari karya sebelumnya pada HTML 4.01, HTML 4.0, HTML 3.2 dan HTML 2.0. Dengan kekayaan fitur, XHTML 1.0 adalah reformulasi dari HTML 4.01 di XML, dan menggabungkan kekuatan HTML 4 dengan kekuatan XML. XHTML 1.0 adalah perubahan besar pertama untuk HTML sejak HTML 4.0 dirilis pada tahun 1997. Ini membawa kekakuan XML ke halaman web dan merupakan kunci dalam pekerjaan W3C untuk menciptakan standar yang menyediakan halaman web lebih kaya pada kisaran yang semakin meningkat dari platform browser termasuk ponsel, televisi, mobil, komunikator, kios dan desktop.

Langkah awal dari XHTML 1.0 adalah merumuskan HTML sebagai aplikasi XML sehingga lebih mudah untuk memproses dan lebih mudah untuk perbaikan. XHTML 1.0 mengambil elemen dan atribut dari pekerjaan sebelumnya W3C tentang HTML 4, dan dapat diinterpretasikan oleh browser yang ada, dengan mengikuti beberapa aturan sederhana. XHTML 1.0 dikembangkan dengan tiga versi. Untuk penentuan versi apa yang akan digunakan hanya dengan menyisipkan baris pada awal dokumen. Misalnya, HTML untuk dokumen ini dimulai dengan garis yang mengatakan bahwa itu adalah menggunakan XHTML 1.0 Strict. Jadi, jika ingin memvalidasi dokumen maka alat yang digunakan akan tahu versi apa yang diterapkan. Masing-masing versi memiliki DTD yang berbeda untuk menetapkan peraturan dan definisi HTML. Versi dari XHTML 1.0 adalah:

- **XHTML 1.0 Strict**

  Digunakan bila menginginkan adanya struktur mark-up yang benar-benar bersih, bebas dari markup yang berhubungan dengan tata letak. Versi ini dapat

digunakan bersama-sama dengan *Cascading Style Sheet* (CSS) untuk mendapatkan efek font, warna, dan tata letak yang sesuai.

- **XHTML 1.0 Transitional**

  Banyak orang menulis halaman Web untuk masyarakat umum untuk mengakses mungkin ingin menggunakan versi XHTML 1.0. Idenya adalah untuk mengambil keuntungan dari fitur XHTML termasuk style sheet tapi tetap membuat penyesuaian kecil untuk markup demi kepentingan orang-orang yang melihat halaman dengan browser lama yang tidak dapat memahami style sheet. Ini termasuk menggunakan body elemen dengan bgcolor, text , dan link atribut.

- **XHTML 1.0 Frameset**

  Digunakan bila ingin menggunakan Frames untuk partisi jendela browser menjadi dua atau lebih frame.

Catatan :
Tulisan lengkap mengenai XHTML 1.0 tersedia dalam bahasa Inggris dengan format HTML, PostScript dan PDF pada (http://www.w3.org/TR/xhtml1)

XHTML Basic

XHTML Basic adalah rekomendasi kedua untuk spesifikasi XHTML. Jenis dokumen XHTML Basic meliputi seperangkat minimal modul yang dibutuhkan untuk menjadi *Host Language Document Type* tambahan termasuk gambar, bentuk, tabel dasar, dan dukungan objek. Hal ini dirancang untuk *Web klien* yang tidak mendukung set lengkap fitur XHTML, misalnya, Web klien seperti ponsel, PDA dan pager. XHTML Basic dirancang sebagai dasar umum yang dapat diperluas. Sebagai contoh, sebuah modul event yang lebih umum dibandingkan dengan event yang terdapat pada

HTML 4, dimana dapat ditambahkan atau dapat diperluas oleh modul tambahan dari Modularisasi XHTML seperti Modul Scripting. Tujuan dari XHTML Basic adalah untuk melayani sebagai bahasa umum didukung oleh berbagai jenis agen pengguna.

Catatan:

Tulisan lengkap mengenai XHTML 1.0 tersedia dalam bahasa Inggris dengan format HTML, PostScript dan PDF pada (http://www.w3.org/TR/xhtml-basic)

XHTML Modularization

XHTML Modularization adalah rekomendasi ketiga dalam rangkaian spesifikasi XHTML. Rekomendasi ini tidak menentukan bahasa markup tapi abstrak modularisasi XHTML dan abstraksi penggunaan Definisi Tipe Dokumen (DTD) XML dan dalam skema XML (versi 1.1). Modularisasi ini menyediakan sarana untuk membuat subset dan memperluas XHTML. Modularisasi XHTML akan memudahkan untuk menggabungkan dengan tag markup untuk hal-hal seperti grafis vektor, multimedia, matematika, perdagangan elektronik dan banyak lagi. Penyedia konten akan lebih mudah untuk memproduksi konten bagi berbagai platform, dengan jaminan yang lebih baik tentang bagaimana konten tersebut diberikan, dan keabsahan isi konten.Desain modular mencerminkan kesadaran bahwa satu ukuran cocok untuk semua perangkat dan pada sistem operasi yang berbeda. Dengan pendekatan ini maka konten yang dibuat tidak lagi bergantung pada kemampuan browser (browser sangat bervariasi dalam hal kemampuan). Sebagai contoh dimana sebuah browser di ponsel tidak bisa menampilkan ukuran yang sama dengan mesin desktop. Ponsel ini bahkan tidak memiliki memori untuk memuat halaman yang dirancang untuk browser desktop.

Catatan:
Tulisan lengkap mengenai XHTML 1.0 tersedia dalam bahasa Inggris dengan format HTML, PostScript dan PDF pada (http://www.w3.org/MarkUp/modularization)

XHTML 1.1 - Modul berbasis XHTML

Rekomendasi ini mendefinisikan jenis dokumen XHTML baru yang didasarkan pada kerangka modul dan modul didefinisikan dalam Modularisasi XHTML. Tujuannya antara untuk memberikan konsistesi, jenis dokumen yang rapi dan dipisahkan dari fungsi. Jenis dokumen ini pada dasarnya adalah reformulasi dari XHTML 1.0 Strict dengan menggunakan Modul XHTML. Ini berarti bahwa fasilitas pada jenis dokumen ini tidak mendukung jenis XHTML yang lain, misalnya *XHTML Frames.*

XHTML-Print

XHTML-Print adalah anggota dari keluarga Bahasa XHTML yang didefinisikan oleh *Modularisasi XHTML.* Dokumen jenis ini dirancang untuk mencetak dari perangkat mobile demi menghemat biaya cetakan, dimana printernya sendiri mungkin tidak memiliki buffer halaman penuh karena pada umumnya proses mencetak dari atas-ke-bawah dan kiri-ke-kanan dengan kertas dalam orientasi potret. XHTML-Print juga ditargetkan mencetak dalam lingkungan di mana tidak layak atau diinginkan untuk menginstal driver printer-spesifik dan mengakomodasi perbedaan pada format output.

4.2.1.2 Perbedaan HTML dan XHTML

HTML ditulis dalam sintaks yang unik didefinisikan oleh SGML, sedangkan XHTML ditulis dalam sintaks SGML dan didefinisikan berbeda yang disebut XML. Keduanya sebenarnya hampir sama dikarenakan XHTML adalah warisan atau berasal dari HTML yang dirancang untuk menyesuaikan standar XML.

Ada beberapa perbedaan antara kedua bahasa markup ini, antara lain:

- Dokumen jenis XHTML dapat dijalankan pada perangkat mobile seperti ponsel, pager dan lain-lain sehingga tidak semata-mata hanya pada computer PC.
- Penggunaan penulisan elemen dan attribut pada dokumen XHTML bersifat *sensitive case* (huruf kecil) yang hal ini jauh berbeda dengan dokumen HTML.
- Penulisan tag pada XHTML selalu berpasangan, dan hal ini sedikit berbeda dengan HTML dimana sebagian tag elemennya tidak harus di tutup, contohnya adalah elemen <br>.
- Dalam dunia XHTML, segala tag harus bersarang secara teratur (properly nested), artinya bahwa jika anda membuka tag "a" setelah itu membuka tag baru di dalamnya, maka tag yang paling baru harus ditutup duluan dan tag yang terbuka pertama harus ditutup paling akhir. Walaupun nesting ini juga terdapat dalam dunia HTML namun tidak seketat peraturan di XHTML.
- Tiap value pada attribute harus terbungkus dengan tanda kutip ganda atau tunggal dan atributnya sendiri tidak boleh disingkat.
- Image tag harus terdapat alt attribute yang menyediakan deskripsi image, untuk memungkinkan mereka memiliki beberapa persyaratan untuk aksesibilitas bersama dengan standar web yang berbeda.
- Persyaratan lain dari XHTML adalah adanya pernyataan dari DOCTYPE yang menentukan aturan mana yang diikuti oleh dokumen anda (aturan yang diwarisi dari XML). Pernyataan ini akan ditemukan pada barisan pertama dari atas ketika anda mengaktifkan halaman source kode XHTML yaitu tipe dokumen deklarasi (juga disebut dengan DTD atau DOCTYPE). Sebagian besar

halaman web yang diciptakan hari ini akan menyertakan deklarasi DOCTYPE tersebut. Ada 3 DTD untuk XHTML: Strict (hanya akan validasi jika tanpa tag usang), Transisi (masih akan memvalidasi dengan tag usang), dan Frameset (untuk halaman yang "set up frame"). Semua dokumen XHTML harus sesuai dengan aturan sintaks XML.

4.2.2 XML

XML kependekan dari *eXtensible Markup Language*, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML sebenarnya bukan teknologi baru, tapi merupakan turunan dari SGML yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan markup language yang tidak kalah hebatnya dengan SGML.

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup(diawali dengan '</ 'diakhiri '>') dan atribut elemen(parameter yang dinyatakan dalam tag pembuka misal <form name="isidata">). Hanya bedanya, HTML medefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak kita. Untuk lebih jelasnya lihat contah dibawah:

```
<pesan>
   <dari>MIS Manager</dari>
   <buat>HRD Manager</buat>
   <buat>Bagian rekrut</buat>
```

<buat>Computer Suport team</buat>
<subyek>Permohonan Tenaga kerja baru</subyek>
<isi>Mohon diberikan tenaga kerja baru untuk mengisi lowongan di Departemen MIS</isi>
</pesan>

pada contoh diatas <pesan>, <dari> <buat>,dan <isi> bukanlah tag standard yang telah di tetapkan dalam XML. Tag-tag itu kita buat sendiri sesuai keinginan kita. Sampai di sini XML tidak melakukan apapun. Yang ada hanyalah informasi yang di kemas dengan tag-tag XML. Kita harus membuat software lagi untuk untuk mengirim, menerima atau menampilkan informasi di dalamnya.

XML untuk saat ini bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri. XML terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standard layaknya heading, paragraph, table dan lain sebagainya.

Sama dengan HTML, file XML berbentuk teks sehingga bila diperlukan kita bisa membacanya tanpa memerlukan bantuan software khusus. Hal ini memudahkan pengembang aplikasi yang menggunakan XML untuk mendebug programnya. XML lebih fleksibel dibanding HTML dalam hal kemampuannya menyimpan informasi dan data. Pada XML kita bisa menyimpan data baik dalam atribut maupun sebagai isi elemen yang diletakkan diantara tag pembuka dan tag penutup. Kelebihan lain yang dimiliki XML adalah bahwa informasi bisa di pertukarkan dari satu system ke system lain yang berbeda platform. Misalnya dari Windows ke Unix, atau dari PC ke Machintosh bahkan dari internet ke handphone dengan teknologi WAP.

4.2.2.1 Bagian-Bagian dari Dokumen XML

Sebuah dokumen XML terdiri dari bagian bagian yang disebut dengan node. Node-node itu adalah:

a. **Root node** yaitu node yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu root node. Node-node yang lainnya berada di dalam root node.

b. **Element node** yaitu bagian dari dokumen XML yang ditandai dengan tag pembuka dan tag penutup, atau bisa juga sebuah tag tunggal elemen kosong seperti <anggota nama="budi"/> . Root node biasa juga disebut root element.

c. **Attribute note** termasuk nama dan nilai atribut ditulis pada tag awal sebuah elemen atau pada tag tunggal.

d. **Text node**, adalah text yang merupakan isi dari sebuah elemen, ditulis diantara tag pembuka dan tag penutup.

e. **Comment node** adalah baris yang tidak dieksekusi oleh parser.

f. **Processing Instruction node**, adalah perintah pengolahan dalam dokumen XML. Node ini ditandai awali dengan karakter <? Dan diakhiri dengan ?>. Tapi perlu diingat bahwa header standard XML <?xml version="1.0" encoding="iso-8859-1"?> bukanlah processing instruction node. Header standard bukanlah bagian dari hirarki pohon dokumen XML.

g. **NameSpace Node**, node ini mewakili deklarasi namespace

Beberapa hal utama yang perlu diperhatikan dalam penulisan script dengan menggunakan XML, yaitu:

- Nilai atribut harus diletakkan diantara tanda petik
- Seperti HTML, XML memiliki atribut. Nilai atribut harus

diletakkan diantara dua tanda petik. Tidak masalah apakah tanda petik tunggal atau tanda petik ganda. Contoh dibawah ini dua-duanya benar

<pesan dari="lusy"> atau

<pesan dari='lusy'>

- Penamaan tag dan atribut

  Nama tag bisa terdiri dari huruf, angka dan underscore("_"). Karakter awal nama tag harus berupa huruf atau underscore ("_"), tidak diawali dengan kata xml atau XML, (misal:<xmlstring>), dan tidak mengandung spasi. Aturan penamaan atribut sama dengan aturan penamaan tag.

- Menyisipkan komentar

  Pada bahasa pemrograman atau scripting kita mengenal adanya komentar (comment). Komentar adalah kalimat/ baris yang tidak dieksekusi oleh compiler, browser atau parser. Untuk menyisipkan komentar pada dokumen XML caranya adalah sebagai berikut:

  <!—Baris ini tidak di eksekusi oleh parser —>

- Menggunakan Karakter Illegal pada XML

  Sama seperti pada HTML, anda tidak bisa menggunakan karakter seperti kurung siku (< atau >), petik tunggal ('), dan petik ganda (") .

  Contoh di bawah ini akan menghasilkan error kalau di eksekusi oleh browser.

  <syarat>jika jumlah < 1000 maka</syarat>

  Untuk menghindarinya, kita harus menggantikannya dengan entity reference seperti di bawah ini:

  <syarat>jika jumlah < 1000 maka</syarat>

  Selengkapnya perhatikan tabel di bawah ini:

Tabel 5.1 *Karakter pada XML*

Entity references	Character	Character Name
<	<	Less Than
>	>	Greater then
&	&	Ampersand
'	'	Apostrophe
"	"	Quotation mark

Document Type Definition (DTD)

Sesuai namanya DTD berfungsi untuk mendefinisikan tipe documen XML. Pada saat mempelajari salah satu bahasa pemrograman atau scripting anda diperkenalkan dengan deklarasi variable, deklarasi fungsi dan deklarasi tipe data. Serupa dengan itu, DTD mendefinisikan struktur dokumen XML dengan daftar element yang digunakan.

Contoh di bawah ini akan memperjelas pengertian DTD

```
<?xml version ='1.0' encoding='utf-8'?>
<!DOCTYPE organisasi [
<!ELEMENT organisasi (anggota)>
<!ELEMENT anggota (nama,alamat,kelamin,jabatan)>
<!ELEMENT nama (#PCDATA)>
<!ELEMENT alamat (#PCDATA)>
<!ELEMENT kelamin (#PCDATA)>
<!ELEMENT jabatan (#PCDATA)>
<!ENTITY org "Forum Komunikasi Remaja Masjid se-Jabotabek"> ]>
<organisasi nama = "&org;">
    <anggota>
```

```
        <nama>Budi Hermanto</nama>
        <alamat>Kebayoran</alamat>
        <kelamin>laki-laki</kelamin>
        <jabatan>ketua</jabatan>
    </anggota>
    </organisasi>
```

Contoh di atas adalah DTD yang menjadi satu dalam satu dokumen dengan XML. Kita bisa memisahkan DTD pada file tersendiri, terpisah dari dokumen XML-nya. Caranya, perhatikan contoh berikut:

```
<?xml version ='1.0' encoding='utf-8'?>
<!DOCTYPE organisasi SYSTEM "organisasi.dtd">
<organisasi nama = "&org;">
<anggota>
    <nama>Budi Hermanto</nama>
    <alamat>Kebayoran</alamat>
    <kelamin>laki-laki</kelamin>
    <jabatan>ketua</jabatan>
    </anggota>
</organisasi>
```

Lalu Anda membuat file baru dengan nama "organisasi.dtd" yang berisi deklarasi DTD seperti contoh berikut:

```
<!ELEMENT organisasi (anggota)>
<!ELEMENT anggota (nama,alamat,kelamin,jabatan)>
<!ELEMENT nama (#PCDATA)>
<!ELEMENT alamat (#PCDATA)>
<!ELEMENT kelamin (#PCDATA)>
<!ELEMENT jabatan (#PCDATA)>
<!ENTITY org "Forum Komunikasi Remaja Masjid se-Jabotabek">
```

DTD memungkinkan format yang unik untuk setiap file xml. DTD akan sangat berguna bila kita membuat aplikasi dalam Visual Basic, ASP atau bahasa pemrograman lain yang mendukung XML, yaitu untuk memastikan bahwa data yang diterima aplikasi itu adalah data yang valid. Atau bermanfaat juga digunakan bila satu organisasi menyepakati penggunaan satu DTD untuk tukar menukar data dan informasi.

Unsur-unsur yang dideklarasikan dalam DTD adalah semua unsur yang membentuk suatu dokumen XML yaitu:

- **Element**, satu blok data yang diawali tag pembuka dan tag penutup.
- **Attribute**, informasi pendukung element yang disertakan pada tag pembuka
- **Entity**, karakter pengganti untuk sekumpulan informasi yang didefinisikan

Element

Bila sebuah element mengandung beberapa child element, maka kita perlu mendeklarasikan child element apa saja yang dipunyai element tersebut. Pada contoh diatas kita melihat deklarasi element <!ELEMENT organisasi (anggota)>. <!ELEMENT anggota (nama, alamat, kelamin, jabatan)> Maksudnya adalah element bernama organisasi memiliki satu child element bernama anggota. Lalu element bernama anggota itu sendiri mempunyai empat child element yang bernama nama, alamat, kelamin dan jabatan. Setelah itu kita perlu medeklarasikan juga type dari element-element di atas.

```
<!ELEMENT nama (#PCDATA)>
<!ELEMENT alamat (#PCDATA)>
<!ELEMENT kelamin (#PCDATA)>
<!ELEMENT jabatan (#PCDATA)>
```

Contoh di atas adalah cara untuk mendeklarasikan type elemen, dimana element nama, alamat, kelamin dan jabatan semuanya bertipe (#PCDATA).

Attribute

Agar dokumen XML kita valid, kita juga perlu mendefinisikan semua attribut yang akan kita gunakan dalam dokumen kita. Untuk mendefinisikannya kita akan menggunakan Attribute list declaration. Caranya seperti berikut:

<!ATTLIST namaelemen spesifikasiattribut>

namaelemen adalah nama elemen dimana attribute itu digunakan. Sedangkan spesifikasiattribute adalah serangkaian informasi tentang attribute itu. Unsur yang membentuknya antara lain nama attribut, type attribute, nilai awal (default value), dan sifat attribute.

Contoh:

<!ATTLIST ORGANISASI Nama CDATA #FIXED "HMTE">

Maksud dari contoh di atas adalah elemen ORGANISASI memiliki Attribute Nama yang bertype CDATA. Sifat attribute dalam hal ini adalah #FIXED, yaitu nilai dari attribute nama harus seperti yang dideklarasikan ("HMTE" adalah nilai default yang diberikan bila kita tidak menyebutkannya). Sifat-sifat attribute selengkapnya adalah:

- FIXED, bila attribute kita deklarasikan dengan ini, kita bisa mencantumkan atau menghilangkan attribute pada element bersangkutan. Bila kita menghilangkannya, secara otomatis parser akan mencantumkan attribute dengan nilai sesuai dengan nilai awal yang diberikan.
- REQUIRED, ini menandakan bahwa attribut yang bersangkutan tidak bisa dihilangkan. Pendeklarasiannya tidak membutuhkan nilai awal. Bila anda paksakan, browser akan menampilkan pesan error.

- IMPLIED, bila kita menggunakannya berarti kita bisa membuang atau mencantumkan attribute. Bila kita tidak mencantumkannya, prosesor tidak akan memberikan nilai default.

Entity

Dengan menggunakan entity XML kita bisa menggantikan kalimat yang panjang atau satu blok elemen yang sering kita gunakan dengan sebuah pengenal singkat. Misalnya kita ingin menggantikan kalimat "Manajemen Data dan Informasi dengan XML/XSL" dengan entity &judul;.(kita telah menyinggung pada pembahasan terdahulu bahwa entity diawali dengan & dan diakhiri dengan ;).

Sekali entity didefinisikan di dalam DTD, kita bisa menggunakannya dimana saja pada seluruh dokumen XML.

```
<?xml version="1.0" encoding="iso-8859-1">
<!DOCTYPE organisasi [
<!ENTITY judul "Manajemen data dan informasi dengan XML/XSLT">
]>
    <resensi>
        <buku judul="&judul;">
        <ulasan>Buku yang berjudul &judul; ini ditulis oleh Moh Junaedi..</ulasan>
    </buku>
</resensi>
```

4.2.2.2 Perbedaan HTML dan XML

- Pada HTML beberapa elemen tidak harus berpasangan. Contoh berikut ini diperbolehkan dalam penulisan HTML.

```
<p>paragraph pertama
<p>paragarap kedua
```

yang demikian tidak berlaku pada XML. Kita harus menulis pula tag penutup untuk setiap tag yang kita buat. Penulisannya harus seperti ini

paragraph pertama

paragarap kedua

- Tag tunggal hanya diperbolehkan untuk elemen kosong. Contoh penulisannya sebagai berikut:

  <anggota nama="budi" />

  huruf besar dan kecilnya.

  <contoh>ini penulisan yang salah</Contoh>

  <contoh>ini baru betul</contoh>

- XML mempertahankan spasi seperti apa adanya Berbeda dengan HTML, XML menampilkan spasi persis bagaimana data ditulis.

4.3 Pemilihan Basis Data (*Database*)

4.3.1 Konsep *Database*

Konsep dasar dari basis data (database) adalah kumpulan dari catatan-catatan atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional dimana data didalamnya dapat mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan. Setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminolog matematika). Dalam model ini hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain

seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. Database adalah sebuah struktur data yang umumnya dikategorikan dalam 2 hal:

1. Flat File Database dan
2. Database Relasional

Flat File Database

Flat file database adalah suatu database yang didesain menyertakan suatu tabel tunggal. Flat file database meletakkan seluruh data kedalam tabel tunggal, atau daftar, dengan kolom-kolom yang merepresentasikan seluruh parameter. Sebuah flat file bisa terdiri dari banyak kolom, seringkali dengan duplikasi data yang cenderung menyebabkan kerusakan data *(data corruption)*. Jika ada keputusan untuk menyatukan beberapa flat file, maka Anda harus melakukan copy dan paste informasi yang relevan dari satu file ke file yang lainnya. Dari kasus ini, tidak ada otomatisasi diantara dua flat file. Jika Anda memiliki dua atau lebih flat file yang berisi data alamat klien, sebagai contoh, klien telah berpindah alamat, maka Anda harus merubah secara manual alamat klien tersebut yang ada dalam setiap flat file. Perubahan informasi dalam satu file tidak memiliki sangkut paut dengan file lainnya. Flat file menawarkan fungsionalitas untuk menyimpan informasi, memanipulasi kolom-kolom, mencetak dan menampilkan informasi yang terformat, pertukaran informasi dengan orang lain melalui email dan melalui internet. Beberapa flat file bisa dikerjakan pada file-file eksternal, seperti text editor, agar memperluas fungsionalitas dan mengatur informasi yang berhubungan. Mendesain flat file database adalah sederhana, dan memerlukan sedikit pengetahuan desain database. Flat file dapat dikembangkan dengan hanya menggunakan satu database engine. Flat file dapat dibuat dalam Relasional database engine dengan tidak mengambil manfaat atau keuntungan dari konsep desain relasional.

Database Relasional

Database Relasional adalah database yang menggabungkan tabel-tabel dengan berbagai metode untuk dapat berkerjasama. Hubungan antar tabel data dapat dibandingkan, disatukan, dan ditampilkan dalam form-form database. Sebagian besar database Relasional menawarkan fungsionalitas untuk berbagi (*share*) data. Baik share data melalui jaringan internet, laptop dan perangkat elektronik lainnya seperti palm pilots ataupun dengan software sistem yang lain.

Mendesain suatu database relasional memerlukan perencanaan yang lebih dari pada flat file database. Dengan flat file, menambah informasi sepertinya perlu dipertimbangkan. Dengan database Relasional, dapat menyimpan data ke dalam tabel-tabel sedemikian rupa sehingga hubungan antar tabel dapat dimengerti. Membangun suatu database Relasional sangat tergantung pada kemampuan untuk menetapkan suatu model relasional. Model harus menggambarkan secara penuh bagaimana data diorganisir dalam ketentuan struktur data, intergriti, query, manipulasi, dan penyimpanan. Database relasional memungkinkan user untuk mendefinisikan kolom-kolom record tertentu sebagai key atau index, melakukan pencarian, menghubungkan record-record antar tabel dan menetapkan batasan integrita . Query pencarian lebih cepat dan lebih akurat jika berdas kan nilai yang telah diindex.Batasan integritas dapat diteta kan untuk menjamin bahwa hubungan antar tabel ters but tidak mengalami duplikasi dan redudansi.

Pada database yang memiliki struktur relasi nal terdapat tabel-tabel yang menyimpan data. Setiap tab l terdiri dari kolom dan baris. Sebuah kolom mendefinisikar enis informasi apa yang akan disimpan. Kolom mendefinis an jenis informasi apa yang akan disimpan, sedangkan se ah baris adalah data aktual yang disimpan. Setiap baris ri tabel adalah

masukan dari tabel tersebut dan berisi nilai-nilai untuk setiap kolom tabel tersebut. Database akan menjadi sangat berguna saat diperlukan ketika ingin menyimpan informasi yang dikategorikan secara logis. Contoh, ketika ingin menyimpan informasi tentang kegiatan bisnis pada sebuah perusahaan. Dengan database, data-data kegiatan bisnis pada perusahaan tersebut dapat dibangun dengan merelasikan beberapa tabel diantaranya adalah tabel keryawan, tabel penjualan, tabel pembelian, tabel piutang dan lain-lain sehingga proses pengolahan data pada proses bisnis tersebut dapat digunakan untuk proses pengambilan keputusan.

Kelebihan Database Relasional dibandingkan database flat, antara lain:

a. Menawarkan proses reporting yang lebih baik, dengan berbagai report generator yang memfilter dan menampilkan kolom-kolom pilihan.
b. Relasional database menawarkan kemampuan membuat module-module reporting sesuai dengan kebutuhan pengguna.
c. Menawarkan kemampuan mengimpor dan mengekspor data dari software lainnya.

Terdapat tiga sistem database Relasional yang utama:

1. Proprietary
2. Open Source, dan
3. Embedded.

Database Relasional yang proprietary biasanya memerlukan penggunaan bahasa-bahasa pengembangan yang juga bersifat proprietary guna menyempurnakan SQL. Sebagai contoh MS Access yang menggabungkan visual basic dengan SQL.

Database Open source, seperti MySQL, didistribusikan dengan gratis agar mendorong pengembangan user.

Database Embedded dikemas sebagai bagian dari paket-paket software lainnya, seperti paket-paket software taxpreparation. Vendor mensuplai database, dan seluruh perlengkapan-perlengkapan manipulasi yang berhubungan, untuk mengontrol struktur database. Database ini biasanya menyediakan tools (perangkat) untuk mengaudit bekas-bekas transaksi.

4.3.2 Mysql dan PostgreSql sebagai Database Open Source

Mysql

Mysql adalah sebuah program database server yang bersifat multi user dengan menggunakan perintah dasar sql (*Structured query language*). Mysql pertama kali dirintis oleh seorang programmer database bernama **Michael Widenius.** Proyek pengembangan MySQL menyediakan *source code* yang gratis dibawah kententuan GNU General Public License serta berada dalam perjanjian yang proprietary. MySQL merupakan pilihan populer untuk database yang digunakan pada aplikasi web, dan komponen utama dari LAMP (*Linux Apache MySQL Php/Perl/Phyton).* Sedangkan untuk penggunaan yang bersifat komersil tersedia beberapa edisi khusus dan menawarkan beberapa fungsionalitas tambahan. Aplikasi yang menggunakan database MySQL antara lain TYPO3, Joomla, Wordpress, phpBB, Drupal dan perangkat lunak lain yang dibangun di atas LAMP. MySQL juga digunakan dalam sebuah website dengan skala yang lebih besar seperti Wikipedia, Google (meskipun tidak untuk pencarian), Facebook, Twitter, Flickr, Nokia.com dan Youtube.

Antarmuka pada MySQL (*Interface)*

MySQL merupakan suatu RDBMS dimana pengelolaan data yang terdapat dalam database dapat menggunakan baris perintah (*command line*) atau *front end* dari perangkat lunak desktop atau aplikasi web.

Command Line

Penggunaan *command line*, telah dikembangkan oleh pihak ketiga untuk mengelola server MYSQL. Maatkit adalah sebuah alat yang bersifat cross platform untuk MySQL, PostgreSQL dan Memcached dan dikembangkan di Perl. Maatkit dapat digunakan untuk membuktikan replikasi bekerja dengan benar, memperbaiki data yang rusak, mengotomasi tugas yang berulang dan mempercepat server. Maatkit disertakan dengan GNU pada beberapa distribusi Linux seperti CentOS, Debian, Fedora dan Ubuntu.

Grafis

MySQL Workbench adalah lingkungan terintegrasi bebas yang dikembangkan oleh MySQL AB, yang memungkinkan pengguna untuk mengelola database MySQL berbasis grafis, desain visual struktur database, mengelola desain database dan pemodelan, pengembangan SQL (menggantikan *MySQL Query Browser*), dan administrasi database (menggantikan MySQL administrator).

Aplikasi pihak ketiga yang memungkinkan pengguna untuk bekerja dengan struktur database dan data visual adalah:

- *Adminer*, merupakan *front end* MySQL yang bersifat *free* dan ditulis dengan menggunakan script PHP yang mendukung untuk pengelolaan banyak database dan dilengkapi dengan *skin* CSS. Informasi lengkap mengenai Adminer dapat dilihat pada *www.**adminer**.org*.
- *DaDaBIK*, bersifat *free* dan *open source* yang dapat digunakan untuk menyesuaikan (customize) CRUD (*Create, Read, Update dan Delete*) pada front end MySQL dan ditulis dalam bahasa PHP. Informasi lengkap mengenai *DaDaBIK* dapat dilihat pada *www.**dadabik**.org*.

- *dbForge GUI*, alat yang dapat digunakan untuk manajemen database yang mencakup aplikasi terpisah untuk perbandingan skema dan sinkronisasi, perbandingan data dan sinkronisasi, dan pengembangan query.
- *HeidiSQL*, memiliki fitur yang lengkap sebagai *front end* dan berjalan pada sistem operasi Windows serta dapat terhubung ke server MySQL lokal atau remote untuk mengelola database, tabel, struktur kolom, dan datayang terdapat pada *record*.
- *LibreOffice Base*, memungkinkan untuk pembuatan dan pengelolaan database, penyusunan bentuk dan laporan yang memberikan pengguna akhir akses yang mudah terhadap data. Seperti Access, dapat digunakan sebagai front-end untuk berbagai sistem database, termasuk database Access (JET), sumber data ODBC dan MySQL atau PostgreSQL .
- *Navicat*, aplikasi yang menyediakan manajemen database yang dikembangkan untuk Windows, Macintosh dan Linux.
- *OpenOffice.org - OpenOffice.org Base*, dapat mengelola database MySQL jika seluruh paket diinstal bersifat bebas dan open source.
- *phpMyAdmin*, bersifat *free* untuk *front end* aplikasi berbasis web dan banyak dipasang oleh web host di seluruh dunia. *Phpmyadmin* dikembangkan dalam PHP dan termasuk dalam paket LAMP, MAMP dan WAMP.
- *SQLBuddy*, bersifat free dan front end untuk aplikasi berbasis web dan dikembangkan pada PHP.
- *Sequel Pro, free* dan *open-source* untuk Mac OS X .
- SQLYog, user interface(antarmuka) yang dikembangkan oleh komunitas untuk MySQL.

Mesin Penyimpanan pada MySQL(Storage Engine)

Sebagian besar dari kita menggunakan database MySQL dan mayoritas tidak tahu bagaimana memilih mesin basis data tersebut, apa jenis mesin penyimpanan yang tersedia di mysql dan bagaimana mereka berbeda satu sama lain. Dalam buku ini akan dipaparkan ide singkat tentang mesin penyimpanan, keterbatasan pada mesin penyimpanan dan kapan mesin penyimpanan tersebut digunakan. Salah satu keunggulan terbesar yang terdapat pada MySQL adalah fleksibilitas untuk memilih mesin penyimpanan yang berbeda untuk tabel yang berbeda. Mesin penyimpanan ini bertindak untuk menangani jenis tabel yang berbeda. Jadi mesin penyimpanan MySQL mencakup kegiatan yang menangani transaksi yang aman antar tabel maupun transaksi yang tidak aman pada sejumlah tabel yang banyak.

Untuk melihat dukungan mesin penyimpanan yang terdapat pada server, maka dapat digunakan pernyataan *SHOW ENGINES*. Nilai di kolom *Support* menunjukkan apakah mesin dapat digunakan. Sebuah nilai YES, NO, atau DEFAULT menunjukkan bahwa mesin tersedia, tidak tersedia atau tersedia dan saat ini ditetapkan sebagai mesin penyimpanan default. penggunaannya dalam baris perintah MySQL adalah:

```
mysql> SHOW ENGINES\G
*************************** 1. row ***************************
Engine: MyISAM
Support: DEFAULT
Comment: Default engine as of MySQL 3.23 with great performance
*************************** 2. row ***************************
Engine: MEMORY
Support: YES
Comment: Hash based, stored in memory, useful for temporary tables
*************************** 3. row ***************************
Engine: InnoDB
Support: YES
Comment: Supports transactions, row-level locking, and foreign keys
*************************** 4. row ***************************
```

Engine: BerkeleyDB
Support: NO
Comment: Supports transactions and page-level locking
************************* 5. row *************************
Engine: BLACKHOLE
Support: YES
Comment: /dev/null storage engine (anything you write to it disappears)
************************* 6. row *************************
Engine: EXAMPLE
Support: YES
Comment: Example storage engine
************************* 7. row *************************
Engine: ARCHIVE
Support: YES
Comment: Archive storage engine
************************* 8. row *************************
Engine: CSV
Support: YES
Comment: CSV storage engine
************************* 9. row *************************
Engine: ndbcluster
Support: NO
Comment: Clustered, fault-tolerant, memory-based tables
************************* 10. row *************************
Engine: FEDERATED
Support: YES
Comment: Federated MySQL storage engine
************************* 11. row *************************
Engine: MRG_MYISAM
Support: YES
Comment: Collection of identical MyISAM tables
************************* 12. row *************************
Engine: ISAM
Support: NO
Comment: Obsolete storage engine

Daftar di atas menampilkan daftar lengkap dari mesin database yang tersedia. Ada beberapa cara untuk menentukan mesin penyimpanan yang akan digunakan. Metode yang paling sederhana adalah dengan mengatur jenis mesin standar dalam file konfigurasi MySQL menggunakan

perintah tertentu. Fleksibilitas lebih besar ditawarkan yang memungkinkan pengguna untuk menentukan mesin penyimpanan default yang akan digunakan MySQL, yang paling jelas adalah untuk menentukan jenis mesin saat membuat table, comtohnya adalah:

CREATE TABLE mytable (id int, title char(20)) ENGINE = INNODB

Pengguna juga dapat mengubah mesin penyimpanan yang digunakan dalam table yang ada dengan perintah :

ALTER TABLE mytable ENGINE = MyISAM

Di balik kemudahan yang ditawarkan tersebut di atas, pengguna harus berhati-hati ketika mengubah jenis tabel dengan cara tersebut seperti membuat modifikasi untuk jenis tabel yang tidak mendukung indeks yang sama, tipe *field* atau mungkin ukuran yang menyebabkan pengguna kehilangan data. Jika pengguna menetapkan mesin penyimpanan yang tidak ada dalam database saat ini maka tabel jenis MyISAM (default) dibuat sebagai penggantinya.

Sebelum mengambil keputusan tentang mesin penyimpanan mana yang akan digunakan, pertama-tama pengguna harus berpikir tentang fungsi inti yang berbeda yang disediakan pada tiap-tiap mesin penyimpanan. Kita dapat membagi fungsi inti ini menjadi empat bagian yaitu jenis field dan tipe data, jenis penguncian, pengurutan dan transaksi.

Field dan tipe data(*Field and data type*)

Meskipun semua mesin mendukung tipe data yang umum, yaitu, bilangan bulat, real dan penyimpanan berbasis karakter, akan tetapi tidak semua mesin mendukung tipe filed lain, khususnya BLOB (*Binary Large Object*) atau tipe TEKS. Mesin lainnya mungkin hanya mendukung lebar karakter dan ukuran data yang terbatas.

Keterbatasan ini, secara langsung akan mempengaruhi informasi yang akan disimpan dan juga mungkin memiliki efek yang terkait dengan jenis pencarian yang akan dilakukan oleh pengguna dan bahkan berpengaruh pada proses pengurutan pada informasi tersebut. Pada gilirannya, perbedaan-perbedaan ini dapat mempengaruhi kinerja dan fungsi dari aplikasi yang dibangun.

Penguncian(*Locking*)

Proses penguncian dalam mesin database mendefenisikan bagaimana mengendalikan akses dan update informasi. Ketika sebuah objek dalam database terkunci maka proses lain tidak dapat memodifikasi terutama untuk proses membaca(*read* data) atau sampai update data telah selesai dieksekusi.

Proses penguncian akan mempengaruhi berapa banyak aplikasi yang berbeda dapat memperbarui informasi dalam database, juga dapat mempengaruhi permintaan pada data. Sebagian besar mekanisme penguncian dikhususkan untuk mencegah beberapa proses pemutakhiran data yang sama. Dampak dari proses penguncian ini, akan terlihat secara signifikan ketika terdapat beberapa aplikasi yang menggunakan database yang sama dan melakukan proses penambahan (pernyataan INSERT) dan perubahan data (pernyataan UPDATE) secara bersama-sama pula.

Proses penguncian didukung oleh mesin penyimpanan yang berbeda pada tingkat objek yang berbeda, dan ini mempengaruhi tingkat konkurensi pada akses informasi. Tiga tingkatan yang berbeda yang didukung pada proses penguncian adalah, penguncian tabel, penguncian blok dan penguncian baris. Penguncian tabel paling sering digunakan dan merupakan penguncian yang disediakan di MyISAM. Penguncian tabel ini akan mengunci seluruh tabel selama update dan akan membatasi jumlah aplikasi yang mem-

perbarui tabel. Penguncian level halaman digunakan oleh mesin penyimpanan DB Berkeley dan mengunci data sesuai dengan halaman (8KB) dari informasi yang sedang diupload. Penguncian pada tingkat baris memberikan yang terbaik, hanya baris tertentu dalam tabel yang terkunci, yang berarti bahwa banyak aplikasi dapat memperbarui baris yang berbeda dari tabel yang sama tanpa mempengaruhi proses penguncian.

Catatan:

Mesin penyuimpanan InnoDB sangat mendukung penguncian tingkat baris.

Pengurutan (*Indexing*)

Pengurutan secara dramatis dapat meningkatkan kinerja ketika mencari dan memulihkan data dari database. Mesin penyimpanan yang berbeda menyediakan teknik pengurutan yang berbeda. Beberapa mesin penyimpanan tidak mendukung pengurutan karena mereka menggunakan dasar pengurutan tabel (misalnya dalam mesin MERGE) atau karena metode penyimpanan data yang tidak mendukung pengurutan (misalnya pada mesin FEDERATED atau BLACKHOLE).

Transaksi (*Transaction*)

1Transaksi memberikan kehandalan data selama proses *update* atau *insert* informasi dengan memungkinkan pengguna untuk menambahkan data ke dalam database, tetapi hanya dilakukan ketika data berada pada situasi tertentu dan tahapan dalam pelaksanaan aplikasi telah selesai dengan sukses. Sebagai contoh, jika mentransfer informasi dari satu account ke account lainnya, maka akan menggunakan transaksi untuk memastikan bahwa baik debit dari satu account dan kredit untuk yang lain berhasil diselesaikan. Jika proses mengalami kegagalan maka pengguna dapat mem-

batalkan transaksi dan perubahan akan hilang akan tetapi jika proses sukses, maka akan dikonfirmasikan tentang perubahan yang dilakukan.

Keterbatasan dan Alasan Penggunaan Pada Mesin Penyimpanan MySQL

a. MyISAM

Mesin MyISAM adalah mesin default di instalasi MySQL dan merupakan turunan dari tipe mesin ISAM asli yang didukung dalam versi awal dari sistem MySQL. Mesin memberikan kombinasi terbaik dari kinerja dan fungsionalitas, meskipun tidak memiliki kemampuan transaksi (seperti InnoDB atau BDB) dan menggunakan *table-level locking*.

Keterbatasan

1. Prose pemulihan *crash* dapat menjadi proses yang memakan waktu karena kurangnya informasi pada MyISAM tentang sebuah log transaksi.
2. MyISAM tidak mendukung untuk pembuatan *foreign key constraints*
3. Semua query UPDATE pada tabel yang sama adalah bersifat serial, hal ini mengindikasikan bahwa mereka melakukan satu per satuan waktu. Keterbatasan seperti ini akan Nampak ketika berhadapan dengan aplikasi yang multi-user.
4. MyISAM mendukung INSERT bersamaan hanya dalam kasus-kasus tertentu.
5. Maksimal 64 indeks per baris,

Kapan MyISAM digunakan?

1. Ketika aplikasi menuntut teks yang lengkap pada kemampuan pencarian. Daripada mendorong semua data Anda ke dalam tabel MyISAM untuk mendapatkan pencarian teks lengkap, mungkin layak untuk membagi

dataset menjadi data yang harus diindeks untuk pencarian teks lengkap dan disimpan menggunakan MyISAM selanjutnya data harus disimpan menggunakan transaksional mesin, seperti InnoDB.

2. Aplikasi yang secara efektif mendukung untuk *single-user* dimana ada sedikit kemungkinan untuk dilakukanya query secara bersamaan pada server MySQL.
3. Ketika melakukan pengujian terbatas atau pembangunan di mana kinerja tidak di bawah pengawasan.

b. MERGE

Mesin penyimpanan MERGE juga dikenal sebagai mesin MRG_MyISAM, merupakan kumpulan tabel MyISAM yang identik yang dapat digunakan sebagai satu tabel. Pengguna dapat mengeksekusi query yang mengembalikan hasil dari beberapa tabel seolah-olah mereka hanya satu tabel saja. Setiap tabel yang digabung harus memiliki definisi tabel yang sama. Tabel MERGE sangat efektif jika dilakukan proses logging baik secara langsung atau tidak langsung ke dalam database MySQL dan membuat tabel khusus per hari, minggu atau bulan dan ingin dapat menghasilkan permintaan agregat dari beberapa tabel.

Keterbatasan

1. Pengguna hanya dapat menggunakan tabel MyISAM yang identik untuk tabel MERGE.
2. Pengguna tidak dapat menggunakan sejumlah fitur MyISAM dalam tabel MERGE. Misalnya, pengguna tidak dapat membuat indeks *fulltext* pada tabel MERGE.
3. Jika tabel MERGE adalah bersifat *non-temporary*, maka semua tabel MyISAM yang mendasarinya harus *non-temporary* juga. Jika tabel MERGE bersifat sementara, tabel MyISAM dapat berupa campuran *temporary* dan *non-temporary*.

4. Merge menggunakan deskripsi tabel. Jika 10 klien menggunakan tabel MERGE yang memetakan ke 10 tabel, server menggunakan (10 × 10) + 10 deskriptor file. (10 deskriptor file data untuk masing-masing 10 klien dan 10 deskriptor file indeks dibagi di antara klien.)
5. Pembacaan kunci(key) lebih lambat. Ketika Anda membaca kunci, mesin penyimpanan MERGE perlu mengeluarkan membaca pada semua tabel yang mendasari untuk memeriksa mana yang paling sesuai dengan kunci yang diberikan. Untuk membaca kunci berikutnya, mesin penyimpanan MERGE perlu mencari buffer untuk menemukan kunci berikutnya.

***Kapan sebaiknya MERGE* digunakan?**

1. Ketika menginginkan kemudahan dalam pengaturan tabel log.
2. Ingin mendapatkan proses yang lebih cepat. Pengguna dapat membagi tabel besar menjadi *read-only* yang didasarkan pada beberapa kriteria dan kemudian meletakkan masing-masing tabel pada disk yang berbeda. Sebuah tabel MERGE ini bisa menjadi jauh lebih cepat daripada menggunakan tabel besar.
3. Melakukan pencarian lebih efisien. Jika pengguna tahu persis apa yang dia cari, maka dia dapat mencari di hanya salah satu tabel split untuk beberapa permintaan.
4. Proses perbaikan lebih efisien. Lebih mudah untuk memperbaiki tabel khusus yang dipetakan ke tabel MERGE daripada memperbaiki sebuah tabel besar tunggal.
5. Langsung memetakan banyak tabel menjadi satu. Sebuah tabel MERGE tidak perlu mempertahankan indeks tersendiri karena dapat menggunakan indeks yang terdapat pada masing-masing tabel.

6. Jika terdapat banyak tabel maka pengguna dapat membuat tabel yang besar yang sesuai kebutuhan, hal ini jauh lebih cepat dan menghemat banyak ruang disk.
7. Pengguna dapat membuat sebuah alias atau sinonim untuk tabel MyISAM dengan mendefinisikan tabel MERGE yang dipetakan dengan tabel tunggal.

c. MEMORY

Mesin penyimpanan MEMORY (sebelumnya dikenal sebagai mesin penyimpanan HEAP) menyimpan semua data dalam memori, ketika server MySQL dimatikan maka data yang tersimpan dalam database MEMORY akan hilang. Akan tetapi, format asli tabel disimpan dan ini memungkinkan penguna untuk membuat tabel sementara yang dapat digunakan untuk menyimpan informasi untuk akses cepat tanpa harus menciptakan tabel setiap kali server database dimulai.

Penggunaan jangka panjang dari mesin penyimpanan MEMORY umumnya tidak ide yang baik, karena data bisa begitu mudah hilang. Namun, memberikan kemudahan bagi pengguna pada optimalisasi RAM untuk mendukung database yang sedang dikerjakan, penggunaan tabel berbasis MEMORY adalah cara yang efisien untuk menjalankan query yang kompleks pada set data yang besar.

Cara terbaik untuk menggunakan tabel MEMORY adalah dengan menggunakan pernyataan SELECT untuk memilih data yang lebih besar dalam disk atau tabel dan kemudian sub-analis akan menganalisis informasi untuk elemen spesifik sesuai dengan yang diinginkan.

d. FEDERATED

Mesin penyimpanan FEDERATED (ditambahkan di MySQL 5.03) memungkinkan pengguna untuk mengakses data dari remote database MySQL dan seolah-olah itu adalah

database lokal. Akibatnya, server MySQL bertindak sebagai proxy untuk *remote server*, menggunakan library MySQL untuk terhubung ke remote host, mengeksekusi query dan kemudian memformat data ke dalam format lokal.

Pada dasarnya ini adalah cara kerja pada server bukan klien, untuk mengakses database remote dan dapat menjadi cara yang efektif untuk menggabungkan data dari beberapa host atau menyalin data yang spesifik dari database remote ke tabel lokal tanpa menggunakan data ekspor dan impor.

e. ARCHIVE

Mesin penyimpanan ARCHIVE hanya mendukung pernyataan INSERT dan SELECT dan tidak mendukung sebagian besar tipe field pada MySQL. Informasi yang tersimpan dalam tabel mesin penyimpanan ARCHIVE dikompresi dan tidak dapat diubah dan jadi tabel ARCHIVE yang sempurna untuk menyimpan data log atau informasi yang tidak aktif lagi digunakan (misalnya data penjualan yang telah lama berlalu).

Sementara informasi disimpan sangat efisien, perawatan harus dilakukan ketika mengakses data yang tersimpan dalam tabel ARCHIVE, karena bentuk informasi yang dikompresi, untuk melakukan pemilihan informasi tertentu maka harus membaca seluruh tabel, dan itu juga berarti dekompresi informasi. Hal ini jelas dapat meningkatkan waktu yang dibutuhkan untuk melakukan pencarian kompleks dan retrievals. Jika Anda melakukan sejumlah besar pertanyaan mengenai informasi dalam tabel ini mungkin lebih mudah untuk sementara menyalin data ke yang lain, tidak dikompresi, tipe data seperti MyISAM.

f. CSV

Mesin penyimpanan data tidak dalam format biner, namun dalam bentuk file CSV (*Command Separated Value*). Oleh karena itu, ada keterbatasan untuk data yang disimpan.

Ini bukan metode yang efisien untuk menyimpan volume data yang besar, atau jenis data yang besar seperti BLOB, meskipun jenis tersebut didukung. Namun, karena data tersebut disimpan dalam format CSV itu sangat portabel, file-file CSV yang dihasilkan dapat dengan mudah diimpor ke banyak paket perangkat lunak yang berbeda, termasuk Excel, OpenOffice dan sistem database seperti Access atau FileMaker.

Secara umum, mesin CSV tidak praktis sebagai mesin database umum akan tetapi metode penyimpanan ini paling efektif dan paling mudah untuk pertukaran data. Apa yang membuatnya begitu nyaman adalah bahwa kita dapat menggunakan pernyataan SELECT dan INSERT untuk membuat database, yang pada gilirannya berarti bahwa kita dengan mudah dapat menghasilkan file CSV berdasarkan permintaan data lainnya.

Mesin penyimpanan CSV juga dapat digunakan sebagai cara yang efektif untuk mendapatkan informasi ke MySQL. Di sini, pengguna dapat membuat tabel pertama, shutdown server MySQL, menyalin file CSV yang telah diekspor dari Excel, Access atau database lain, dan Anda kemudian dapat mengimpor data dan menyalinnya ke tabel MyISAM atau InnoDB.

g. Blackhole

Mesin penyimpanan Blackhole tidak menyimpan data dan retrievals selalu mengembalikan set kosong. Fungsi tersebut dapat digunakan dalam desain database terdistribusi dimana data secara otomatis direplikasi, tetapi tidak disimpan secara lokal. Meskipun pengguna dapat membuat tabel dan indeks, semua pernyataan SQL yang akan menambah atau memperbarui informasi ke database dijalankan tanpa benar-benar menulis data apapun dan struktur database dipertahankan.

Meskipun hal ini kelihatannya seperti latihan yang sia-sia, hal ini memungkinkan pengguna untuk menguji struktur database dan melihat definisi tabel tanpa benar-benar menciptakan data apapun.

Pengguna dapat menggunakan fungsi ini untuk memperbarui satu atau lebih *slave* secara langsung tanpa harus menulis data lokal. Ada sejumlah manfaat potensial untuk fungsi ini.

h. ISAM

Mesin penyimpanan ISAM adalah tipe mesin asli MySQL yang tersedia pada versi MySQL sampai 3,23, ketika mesin penyimpan MyISAM diperkenalkan. ISAM memiliki sejumlah keterbatasan yang berbeda yang membuatnya tidak praktis sebagai mesin database. Ini termasuk format penyimpanan, platform yang asli (dan karena itu tidak portabel antara sistem), ukuran tabel maksimum teks hanya 4GB dan terbatas untuk fungsi pencarian text serta indeks juga lebih terbatas. Oleh karena itu MyISAM didukung pada platform yang sama seperti ISAM, dan menyediakan kompatibilitas yang lebih baik, portabilitas dan kinerja.

Pengguna pasti tidak harus menggunakan ISAM untuk database baru, akan tetapi gunakan MyISAM sebagai gantinya.

i. Berkeley DB (BDB)

Mesin DB Berkeley (BDB) didasarkan pada teknologi yang disediakan oleh sistem penyimpanan DB Berkeley dan dikembangkan oleh *Sleepycat software*. BDB adalah mekanisme penyimpanan berbasis hash, dan kunci(key) ke nilai hash yang disimpan sangat efisien. Hal ini membuat pemulihan informasi terutama ketika diakses secara langsung menggunakan kunci yang unik sangat cepat, dan sejauh tercepat dari jenis database yang tersedia. Memulihkan

catatan penuh bahkan lebih cepat jika data yang cukup pendek untuk disimpan dengan kunci yang unik (panjangnya di bawah 1024 byte). BDB juga merupakan salah satu dari hanya dua jenis mesin penyimpanan yang mendukung transaksi.

j. InnoDB

Mesin penyimpanan InnoDB disediakan oleh *Innobase Oy* dan mendukung semua fungsi database. Kunci utama pada sistem InnoDB adalah *caching database* dan mengindeks struktur di mana indeks dan data cache dalam memori disimpan pada disk. Hal ini memungkinkan pemulihan yang sangat cepat, dan bekerja bahkan pada set data yang sangat besar. Dengan mendukung penguncian tingkat baris, pengguna dapat menambahkan data ke tabel InnoDB tanpa penguncian tabel.

Keterbatasan

1. Query yang menghasilkan scan besar tablespace yang sering lambat ketika menggunakan InnoDB.
2. Konsistensi hanya dipertahankan jika sistem operasi yang mendasari dan perangkat keras dapat menjamin flushes buffer. Keterbatasan ini melekat dalam semua sistem manajemen database transaksional.
3. Tabel InnoDB mengkonsumsi sejumlah besar ruang pada disk, hal ini sekarang sebagian besar tidak relevan mengingat ubiquity besar (ratusan beberapa gigabyte) hard disk drive.

Kapan Menggunakan InnoDB ?

1. Aplikasi Anda menuntut dukungan lapisan penyimpanan gagasan transaksi.
2. Anda membutuhkan *crash recovery*.
3. Situs web atau aplikasi yang sebagian besar multi-user.

Database harus berurusan dengan proses UPDATE yang sering dilakukan ke satu tabel. dan pengguna ingin membuat lebih baik menggunakan multi-processing pada hardware yang sama.

k. NDB Cluster

Mesin penyimpanan ini memungkinkan seseorang untuk membuat tabel *cluster*. Itu berarti pengguna memiliki beberapa master, yang semuanya dapat melakukan sisipan, update dan menghapus di tabel yang sama. NDB memiliki *row-level locking*, akan tetapi karena beberapa keterbatasan arsitektur pada server MySQL untuk proses penggabungan eksekusi maka mesin penyimpanan NDB di sarankan untuk digunakan. Untuk alasan itu, yang terbaik digunakan pada NBD adalah untuk pencarian tabel tunggal primary key.

PostgreSQL

PostgreSQL adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut Perjanjian Lisensi BSD. Perangkat lunak ini merupakan salah satu basis data yang paling banyak digunakan saat ini, selain MySQL dan Oracle. PostgreSQL menyediakan fitur yang berguna untuk replikasi basis data. Fitur-fitur yang disediakan PostgreSQL antara lain DB Mirror, PGPool, Slony, PGCluster, dan lain-lain.

PostgreSQL sering hanya disebut **Postgres** adalah *object-relational database management system*(ORDBMS) tersedia untuk banyak platform termasuk Linux, FreeBSD, Solaris, Microsoft Windows dan Mac OS X. Postgres dirilis di bawah Lisensi PostgreSQL, yang merupakan MIT -style license dengan demikian merupakan perangkat lunak yang bersifat *free* dan *open source*. PostgreSQL dikembangkan oleh Grup PostgreSQL Global Development, yang terdiri dari segelintir relawan yang bekerja dan diawasi oleh perusahaan seperti Red Hat dan EnterpriseDB. Postgres pada umumnya menerapkan standar SQL: 2008, sepenuhnya bersifat

transaksional (termasuk semua pernyataan DDL), memiliki tipe data extensible, operator, metode indeks, fungsi, agregat, bahasa prosedural dan memiliki sejumlah besar ekstensi ditulis oleh pihak ketiga.

Untuk memudahkan administrasi basis data pada PostgeSQL, maka PostgeSQL menyediakan alat bantu bagi para pengguna yakni PGAdmin III yang merupakan alat bantu (tools) berbasis Graphical User Interface (GUI) dan PSql To Postgres yang merupakan alat bantu (tools) bebasis command prompt. Bagi pada database administrator yang tidak puas akan alat bantu (tools) yang dibawa PostgreSQL, maka dapat mendownload alat bantu pihak ketiga (third party tools) dari Internet. Ada banyak tools yang dapat didownload, seperti:

- EMS SQL Manager 2005 For PostgreSQL (http://www.pgsqlmanager.com/), software ini bersifat Free, kinerja yang ditawarkan juga terbilang baik dan tampilannya yang sangat user friendly.
- PostgreSQL Maestro (http://www.sqlmaestro.com/), bersifat Trial 30 hari, jika ingin menggunakannya lebih lanjut, maka kita diharuskan membeli software ini.
- Navicat For PostgreSQL (http://pgsqlsupport. navicat. com/), Software ini bersifat Trial 30 hari, jika ingin menggunakannya lebih lanjut, maka kita diharuskan membeli software ini. Tampilannya lebih sederhana untuk pengguna awam. Namun kemampuannya juga tidak kalah dengan software sejenis seperti EMS Manager atau PostgreSQL Maestro.

Catatan:

Tools hanya merupakan alat bantu, yang lebih penting adalah pengetahuan mengenai bagaimana melalukan administrasi pada sebuah DBMS seperti PostgreSQL.

Pengguna PostgreSQL

- Yahoo! untuk analisa prilaku pengguna web, menyimpan 2 petabyte data dan mengklaim sebagai gudang data terbesar. Menggunakan versi PostgreSQL yang dimodifikasi, dengan engine penyimpanan berbasis kolom yang sepenuhnya berbeda.
- MySpace, situs jejaring sosial populer, menggunakan basisdata Aster nCluster untuk gudang data, dibangun diatas PostgreSQL tanpa modifikasi.
- OpenStreetMap, proyek kolaboratif untuk menciptakan peta dunia yang bebas sunting.
- Afilias, register domain untuk .org, .info, dan sebagainya.
- Sony Online multiplayer online game.
- BASF, platform belanja untuk portal agribisnis
- Skype aplikasi VoIP, basisdata pusat bisnis.
- Sun xVM, perangkat lunak virtualisasi dan otomasi datacenter milik Sun.

4.3.3 Perbedaan Mysql dengan PostgreSQL

Perbandingan dalam buku ini menggunakan versi MySQL 3.23.49/4.0.1 dan PostgreSQL 7.2.

1. Tujuan Desain

Dari semula latar belakang dikembangkannya kedua database ini sudah berbeda. MySQL berkembang dari solusi yang dipakai oleh pembuatnya, TcX AB, dalam memroses data untuk aplikasi Web. Fokusnya adalah pada kecepatan.

PostgreSQL berkembang dari riset akademik. Fokus pengembangan PostgreSQL adalah pada fitur OO, reliabilitas, dan dukungan SQL yang mantap.

Namun, seiring kedua produk ini bertambah matang, keduanya semakin banyak memiliki sifat-sifat ini. MySQL versi 4.x misalnya, berjanji menambahkan fitur-fitur yang

sejak lama diidamkan pemakainya: subselect, view, dsb. Sementara PostgreSQL, yang sempat memiliki masalah stabilitas dan skalabilitas di seri awal versi 6.x, juga kini telah amat menarik dari segi kecepatan.

2. Pengembangan

Pengembangan MySQL diatur secara sentral oleh perusahaan komersial di Swedia bernama MySQL AB (sebelumnya TcX AB). Perusahaan ini memperoleh pemasukan utamanya dari menjual layanan support dan konsultasi MySQL.

PostgreSQL dikembangkan secara lebih terdesentralisasi dan merakyat, namun tetap diatur oleh sebuah kelompok online bernama PostgreSQL Development Group.

MySQL dirilis dalam satuan yang lebih sering (sebulan bisa lebih dari satu kali), sementara PostgreSQL sekitar 4–6 bulan sekali.

3. Jumlah Pengguna

Menurut MySQL AB, saat ini jumlah instalasi MySQL sekitar 3 juta.

PostgreSQL sendiri tidak diketahui pasti berapa jumlah penggunanya; kemungkinan masih berada di bawah MySQL karena banyaknya situs Web dan perusahaan webhosting yang hanya menggunakan MySQL. Plus secara keseluruhan popularitas MySQL (trafik milis, tutorial/artikel yang membahas, dsb) lebih besar daripada PostgreSQL. Tapi karena PostgreSQL juga disertakan secara default di distro-distro Linux seperti Red Hat dan SuSE, jumlah penggunanya pun sudah pasti banyak.

4. Arsitektur dan Portabilitas

MySQL memiliki arsitektur multithreading, sementara PostgreSQL multiproses (forking). Ini berarti PostgreSQL potensial memiliki stabilitas yang lebih tinggi, sebab satu proses anak yang mati tidak akan menyebabkan seluruh

daemon mati—meskipun pada kenyataannya, dulu ini sering terjadi. Di sisi lain, arsitektur dengan forking ini sulit diterapkan ke Windows, sebab Windows amat thread-oriented. Karena itulah, baru MySQL yang memiliki port natif ke Windows. PostgreSQL sendiri saat ini bisa dijalankan di Windows, tapi melalui lapisan emulasi Cygwin.

5. ACID compliance

Sampai sekarang masih banyak yang bilang MySQL itu tidak ACID-compliant. Padahal sejak 2 tahun lalu MySQL sudah mempunyai handler tabel BerkeleyDB, dan belakangan ini InnoDB, sehingga MySQL sudah mendukung transaksi. Handler tabel MySQL yang lama, ISAM dan MyISAM, tidak ACID-compliant. PostgreSQL sendiri sejak lama telah ACID-compliant.

6. Lisensi

Lisensi PostgreSQL lebih liberal. Inilah sebabnya ada banyak produk closed-source dan komersial yang bisa dikembangkan dari source code PostgreSQL. MySQL, karena dilisensi di bawah GPL, tidak boleh dimodifikasi menghasilkan produk turunan yang closed-source.

7. Kecepatan

Soal kecepatan ini relatif dan kadang juga jadi isu sensitif. Baik kedua pihak, maupun pihak ketiga, pernah menerbitkan benchmark yang lalu ditepis atau dicibir karena tidak objektif. Pada dasarnya perbandingan kecepatan keduanya seperti ini: MySQL terkenal cepat dalam melakukan query sederhana. Dengan kata lain, dapat memroses lebih banyak SQL per satuan waktu. Tapi dalam kondisi load tinggi (jumlah koneksi simultan besar), PostgreSQL sering mengalahkan MySQL untuk query dengan klausa JOIN yang kompleks, seperti dialami Tim Perdue saat mencoba kedua database untuk diimplementasikan di SourceForge.net.

Penyebab utamanya adalah karena MySQL menggunakan locking level table dalam UPDATE, sehingga koneksi yang lain tidak bisa membaca table yang bersangkutan sama sekali. Locking inilah juga sebabnya mengapa pada banyak benchmark, MySQL menunjukkan penurunan kinerja yang cukup drastis untuk kondisi jumlah klien simultan tinggi. PostgreSQL mendukung locking di level yang lebih rendah, yaitu row. Table handler baru di MySQL, InnoDB, juga mendukung row level locking. Benchmark InnoDB pada jumlah koneksi tinggi menunjukkan hasil yang cukup menjanjikan (www. innodb. com/bench.html).

Masalah locking tabel bisa diakali dengan membelah-belah tabel, agar satu kelompok row dapat dilock tanpa mengganggu kelompok row lain. Bahkan ada pengguna MySQL yang membelah sebuah tabel besar berisi jutaan record menjadi ribuan tabel kecil-kecil.

8. Stabilitas

Keduanya sudah bisa dibilang cukup hingga amat stabil. Tapi perlu diingat bahwa database manapun—bahkan Oracle—sesekali dapat menyebabkan kerusakan data. Karena itu backup/history periodik dan incremental tetap diperlukan.

9. Fungsi Built-In

MySQL terkenal kaya fungsi built-in, seperti modifikasi string (REPLACE, RIGHT, LTRIM, LCASE), matematika (LOG, LOG10), tanggal, dsb. Dalam hal ini MySQL lebih unggul.

10. Interface

Keduanya sudah amat solid. Mulai dari API C/C++, driver database Perl/Python/PHP/Tcl, ODBC, JDBC telah didukung. User tidak akan kesulitan menggunakan database ini dari berbagai sistem dan bahasa pemrograman. MySQL juga mendukung OLEDB dan memiliki versi embedded untuk dilink bersama aplikasi buatan user sendiri.

11. Full Text Indexing

MySQL mendukung indeks full text secara natif. PostgreSQL mendukung full text searching lewat program lain (contohnya: OpenFTS, openfts.sourceforge.net) yang memanfaatkan tipe data arraynya untuk menyimpan indeks dokumen. Secara umum dapat dikatakan bahwa indexing dengan MySQL lebih praktis, tapi dengan program ketiga lebih banyak fitur dan opsi yang bisa diatur (mis: stemming, parsing kata non-Inggris, dsb). MySQL juga, tentu saja, dapat dipakai sebagai backend bagi program search eksternal (contoh: DBIx::KwIndex, search.cpan.org/search?dist=DBIx-KwIndex), meski mungkin tidak seefisien dibandingkan array di PostgreSQL.

12. Replikasi

Keduanya sudah memiliki replikasi, meski replikasi di MySQL barulah satu arah. Replikasi di PostgreSQL sendiri belum disertakan dalam distribusi standarnya, namun Anda dapat mengunjungi situs gborg.postgresql.org/project/pgreplication/ (proyek pgreplication).

13. Manajemen User dan Keamanan

Kedua database menyimpan informasi user di sebuah database khusus. Sistem perizinan MySQL lebih mendetil daripada PostgreSQL. Misalnya, kita dapat mengeset agar user tertentu yang datang dari host tertentu hanya bisa membaca tabel saja tanpa bisa UPDATE. Di PostgreSQL ini masih bisa dilakukan dengan VIEW misalnya.

Untuk masalah enkripsi koneksi, keduanya mendukung SSL. Ada ekstensi PKIX bagi PostgreSQL yang menarik, sebab dapat membuat tabel terenkripsi: http://www.dimensional.com/~bgiles/pkixdoc/.

14. Tool Web/GUI

MySQL AB mengklaim lebih banyak tool grafis/web yang tersedia untuk MySQL, dan ini nampaknya cukup benar.

15. Tipe Data

PostgreSQL lebih kaya dalam hal tipe data (terutama yang domain-specific seperti tipe data geometris dan MONEY), tapi MySQL sudah mendukung semua tipe data umum.

Di PostgreSQL sebelum 7.1, masih ada keterbatasan yang cukup menyesakkan yaitu ukuran data BLOB maksimum adalah 8-32KB. Sejak 7.1, PostgreSQL juga dapat menyimpan data BLOB besar.

CHAR dan VARCHAR di PostgreSQL dapat menampung hingga 8 juta karakter (bandingkan dengan MySQL yang hanya 255).

16. Modifikasi Tabel

MySQL lebih fleksibel dalam ALTER TABLE. PostgreSQL sendiri terbatas hanya bisa melakukan penambahan kolom, penggantian nama kolom, dan penggantian nama tabel. MySQL mendukung penambahan/penghapusan kolom, penggantian definisi kolom dan sebagainya.

17. Fitur *Object Oriented* dan SQL

Dalam waktu beberapa tahun PostgreSQL akan tetap memiliki fitur yang lebih lengkap dibandingkan MySQL. Lebih banyak fitur dari standar SQL92 yang diimplementasi oleh PostgreSQL. MySQL bahkan belum mendukung subselect. View, trigger, foreign key checking (meski ini sudah ada di InnoDB) dan stored procedure semua hanya ada di PostgreSQL. Sebagai pengembang yang memutuskan memilih salah satu database, user perlu menanyakan kepada diri sendiri dulu apakah ingin lebih bersusah-payah melakukan code around fasilitas-fasilitas yang tidak ada di MySQL tersebut melalui bahasa pemrograman (misalnya, stored procedure diganti dengan user-defined function, subselect diganti beberapa kali SQL yang dibungkus dengan locking, dan tidak

ada "trigger" berarti Anda harus melakukan pengecekan secara manual). Jika tidak ingin repot, lebih baik memilih PostgreSQL. Tapi jika tidak butuh fitur SQL yang rumit-rumit, Anda masih punya kebebasan memilih satu dari dua.

Di samping itu MySQL pun tidak memiliki fitur OO seperti pewarisan tabel dan tipe data, atau tipe data array yang kadang praktis untuk menyimpan banyak item data di dalam satu record.

18. Fitur Unik

MySQL memiliki arsitektur yang memungkinkan sebuah database terdiri dari beberapa jenis tabel, misalnya: yang transaksional dan tidak, yang berbasis di memori atau di disk, yang terkompresi dan yang read-only. MySQL mendukung protokol terkompresi yang bisa menghemat bandwidth dan mengurangi latensi.

PostgreSQL memiliki tipe data array, pewarisan tabel dan tipe data, serta sistem rule. PostgreSQL memiliki tipe-tipe data "antik." Di PostgreSQL Anda dapat menulis stored procedure (atau procedural language, istilah di PostgreSQL) dalam beberapa bahasa: PL/Perl, PL/Tcl, atau PL/PgSQL. PostgreSQL mendukung set/himpunan.

Dari beberapa penjelasan di atas, user dapat menentukan pilihan database yang digunakan untuk implementasi pengembangan aplikasi berbasis web.

4.3.4 SQL

SQL (dibaca "ess-que-el") merupakan kependekan dari *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah Database. Sesuai dengan ANSI, SQL merupakan bahasa standar untuk sistem menajemen database relasional. Statemen SQL memungkinkan user untuk berkomunikasi dengan database seperti proses membaca, menulis, dan memperoleh informasi yang

berguna dari database. Meskipun sifatnya nonprocedural, lebih mudah bekerja dengan SQL daripada dengan kebanyakan bahasa pemrograman seperti PHP, PERL, Java dan lain-lain,

Perintah atau *statement* SQL yang paling sederhana yang memungkinkan seorang user dapat menampilkan atau memperoleh data dari suatu tabel adalah perintah atau statement SELECT. Sesuai dengan namanya, dengan perintah SELECT seorang user dapat memilih data yang spesifik dari tabel untuk menampilkannya. SQL telah distandarisasi, dan versi saat ini mengacu pada SQL92.

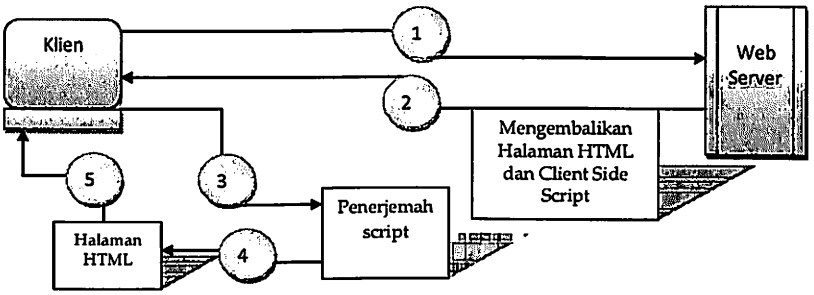
Beberapa database yang mendukung SQL seharusnya menyesuaikan dengan standard SQL saat ini. Standarisasi SQL telah mejadikan SQL sebagai perangkat atau *tool* istimewa yang digunakan dalam pengembangan dan desain web. Sebagian besar program atau software untuk pengembangan aplikasi web, terutama Allaire's Cold Fusion dan Macromedia Dreamweaver Ultradev, mengandalkan pada SQL atau perintah-perintah SQL untuk menghubungkan dan memperoleh informasi dari database.

Untuk informasi lebih detil tentang SQL92, kunjungi:
http://developer.mimer.se/documentation/html_82/Mimer_SQL_Reference_Manual/Intro_SQL_Stds3.html

4.4 Penggunaan Bahasa Pemrograman

4.4.1 *Client Side Scripting*

Client-side scripting umumnya mengacu pada kelas program komputer pada web yang dijalankan pada *sisi klien* oleh pengguna web browser, bukan *server-side* (pada web server). Jenis pemrograman komputer adalah bagian penting dari konsep *Dynamic HTML*(DHTML) yang memungkinkan halaman web untuk ditulis, yaitu untuk memiliki konten yang berbeda dan berubah tergantung pada input pengguna, kondisi lingkungan (seperti waktu hari) atau variabel lain.



Gambar 5.1

Transaksi Klien dan Server pada client side scripting

Client-side script sering tertanam pada dokumen HTML atau XHTML (maka dikenal sebagai *embedded script*), akan tetapi berkas script ini juga dapat ditulis secara terpisah yang akan dirujuk oleh dokumen yang menggunakannya dan dikenal dengan istilah script eksternal. Client-side script memiliki akses lebih besar ke informasi dan fungsi yang tersedia pada browser pengguna dibandingkan server-side script memiliki akses lebih besar ke informasi dan fungsi yang tersedia pada server. Client-side script tidak memerlukan software tambahan pada sisi server, hal inimembuat mereka populer dengan penulis yang tidak memiliki akses administratif ke server. Namun penggunaan bahasa pemrograman jenis ini memang mengharuskan web browser untuk mengerti bahasa scripting yang ditulis oleh programmer. Penggunaan Client-side script terkadang berbenturan dengan batasan keamanan, Client-side script mungkin tidak diizinkan untuk mengakses komputer pengguna di luar aplikasi yang menggunakan web browser. Teknik seperti ActiveX kontrol dapat digunakan untuk menghindari pembatasan ini. Client-side scripting mendorong pengguna untuk selalu menjaga web browser mereka up-to-date untuk menghindari mengekspos komputer mereka dan data untuk celah keamanan yang ditemukan. Script juga dapat digunakan untuk menyederhanakan authoring dokumen yang

aktif. Tindakan pada suatu objek yang dimasukkan ke dalam dokumen HTML dapat disesuaikan oleh script sehingga menghasilkan respon atas perilaku objek tersebut. Hal ini memungkinkan penulis untuk membuat konten web yang menarik dan interaktif.

Elemen Script

<! – SCRIPT adalah elemen yang memuat karakter yang dapat ditempatkan pada HEAD atau BODY sebuah dokumen –>

<!ELEMENT script - - CDATA>

<!ATTLIST script

type	*CDATA*
language	*CDATA*
src	*%URL*

>

type :

Menentukan jenis bahasa scripting, misalnya:

type = "text / javascript" atau

type = "text / vbscript".

language :

Digunakan untuk identifikasi nama bahasa scripting misalnya *"JavaScript"* atau *"VBScript"*.

src :

Atribut src bersifat optional karena akan memberikan URL untuk script eksternal.

Catatan:

Jika atribut src ada maka isi dari elemen SCRIPT harus diabaikan.

Dokumen HTML dapat mencakup unsur SCRIPT ganda, yang dapat ditempatkan dalam dokumen baik pada elemen

head maupun elemen *body*. Hal ini memungkinkan pernyataan Script untuk formulir ditempatkan dekat ke elemen *form* yang sesuai. Apabila tidak terdapat atribut type atau language pada script maka dapat ditentukan oleh META dibagian elemen *head* dokumen, misalnya:

<META HTTP-EQUIV="Content-Script-Type" CONTENT="text/tcl">

Atribut Content akan menentukan jenis media bahasa scripting yang digunakan dan HTTP-EQUIV adalah string literal "Content-Script-Type". Akan tetapi jika ada beberapa META, maka META yang terakhir yang akan menentukan bahasa scripting.

Nama Objek (*Object Name*) Pada Script

Mesin scripting bertanggung jawab untuk mengikat referensi objek dalam skrip untuk semua objek yang terkait dengan dokumen sehingga mesin script dapat mendukung lebih dari satu bahasa. Hal ini memungkinkan penangan terhadap kejadian (*event*) yang ditulis dalam suatu bahasa dapat diikat dan didefinisikan di tempat lain dan menghindari adanya keterbatasan untuk bahasa tertentu. Beberapa bahasa scripting seperti VBScript menyediakan konvensi bahasa untuk mengikat objek dari kode kejadian menjadi script yang dapat menangani kejadian tersebut. Dalam banyak kasus objek berhubungan dengan elemen HTML seperti kolom formulir (*field form*) yang dapat diidentifikasi berdasarkan *markup* dokumen, misalnya nama tag dan nilai atribut yang ada didalamnya. Atribut ID pada dokumen HTML menghasilkan pengidentifikasian yang unik untuk dokumen tertentu, sedangkan atribut *NAME* untuk mendefinisikan *FIELD* pada elemen FORM yang ruang lingkupnya dibatasi hanya untuk FORM yang terlampir. Sistem Scripting memungkinkan untuk menulis script yang dapat menangani kejadian atas suatu objek yang terkait dengan OBJECT atau elemen IMG. Frame Dokumen

memungkinkan satu dokumen yang akan bersarang dengan dokumen lain sehingga penangan script dapat ditempatkan dalam dokumen induk dan digunakan untuk mengontrol perilaku dokumen anak. Script juga dapat digunakan untuk objek eksternal, seperti agen pengguna atau aplikasi lainnya.

Pengendalian Kejadian (*Event*)

Secara umum sejumlah kejadian dapat ditangani dengan cara menempatkan atribut pada elemen HTML dan berhubungan dengan objek yang menghasilkan *event* tersebut. Nama atribut untuk penangan kejadian bersifat *case insensitive*. Nilai atribut sendiri adalah bahasa scripting dalam bentuk string dan berfungsi untuk memberikan petunjuk dijalankannya scripting ketika terjadinya sebuah kejadian (event) yang terkait. Untuk melihat penggunaan penanganan kejadian pada suatu dokumen, dapat dilihat pada contoh berikut. Dalam contoh berikut, username adalah bidang teks yang diperlukan. Ketika pengguna meninggalkan field, maka kejadian OnBlur akan memanggil fungsi JavaScript demi mengkonfirmasi nilai yang dapat diterima pada field tersebut. Contoh : *<INPUT Name="username" onBlur="validUserName(this.value)">*

Berikut ini adalah contoh lain pada JavaScript:

```
<NAMA INPUT = "num"
   onChange = "if (checkNum (this.value, 1, 10)!)
      {This.focus (); this.select ();} else {TerimaKasih ()} "
   NILAI = "0">
```

Untuk penanganan kejadian tersebut di atas maka atribut script akan didefenisikan sebagai CDATA. Proses pada SGML mensyaratkan 2 hal untuk nilai CDATA yaitu pertama pengganti entitas terjadi dalam nilai atribut dan kedua bahwa nilai atribut akan dibatasi oleh LIT (' ') dan LITA (" "). Syarat kedua mengindikasikan bahwa nilai

atribut pada pembatas literal yang terakhir harus sama dengan yang digunakan untuk memulai nilai atribut.

berikut ini adalah contoh bagaimana penanganan kejadian yang terdefinisi dalam dokument HTML.

```
<!ATTLIST SELECT
    name        CDATA    # Harus Ada
    size        NUMBER
    multiple    (multiple)
    onFocus     CDATA
    onBlur      CDATA
    >
```

Beberapa kejadian yang umum digunakan dalam aplikasi berbasis web adalah sebagai berikut:

onLoad

Kejadian(event) ini dapat dilihat ketika browser selesai loading jendela atau semua frame dalam sebuah FRAMESET. Pengendali event onLoad akan mengeksekusi script ketika terjadi event tersebut. Atribut ini hanya dapat digunakan dengan elemen BODY atau FRAMESET.

onUnload

Sebuah event onUnload akan terjadi ketika keluar dari dokumen. Atribut ini hanya dapat digunakan dengan elemen BODY atau FRAMESET.

onClick

Sebuah peristiwa klik terjadi ketika sebuah *anchor* atau field pada form diklik. Pengendali event onClick mengeksekusi scriptlet ketika peristiwa terjadi klik. Event ini dihasilkan oleh tombol, checkbox, tombol radio, hypertext link, reset dan tombol submit. Atribut ini dapat digunakan hanya oleh elemen INPUT dan elemen anchor.

onMouseOver

Event ini dikirim ketika sebuah *mouse* dipindahkan pada sebuah *anchor*. Atribut ini hanya dapat digunakan oleh elemen anchor dan elemen AREA.

onMouseOut

Event ini dikirim ketika sebuah *mouse* keluar dari *anchor* atau textarea. Atribut ini hanya dapat digunakan oleh elemen anchor dan elemen AREA.

onFocus

Event ini terjadi ketika *pointer* berada pada elemen input, baik menggunakan klik *mouse* ataupun menggunakan fungsi tab. Atribut ini dapat digunakan pada elemen INPUT, SELECT dan elemen TEXTAREA.

onBlur

Sebuah event onBlur terjadi ketika bentuk bidang kehilangan fokus masukan. Atribut ini hanya dapat digunakan pada INPUT, SELECT dan elemen TEXTAREA.

onSubmit

Sebuah event submit terjadi ketika pengguna mengirimkan formulir. Hal ini dapat digunakan untuk mengontrol apakah isi form sebenarnya disampaikan atau tidak. Misalnya, JavaScript tidak akan menyerahkan formulir jika scriptlet untuk event onSubmit kembali palsu. Atribut ini hanya dapat digunakan dengan elemen FORM.

onSelect

Event onSelect terjadi ketika pengguna memilih teks yang berada dalam *field* teks tunggal atau multi-baris. Atribut ini hanya dapat digunakan pada elemen INPUT dan elemen TEXTAREA.

onChange

Sebuah peristiwa onChange terjadi ketika adanya perubahan pada focus atau adanya modifikasi pada nilai yang dimasukkan. Atribut ini hanya dapat digunakan pada SELECT, INPUT dan elemen TEXTAREA.

Untuk menciptakan halaman web yang bersifat interaktif dan dinamis, maka teknologi berbasis JavaScript atau VBScript dapat diambil salah satunya. Kedua teknologi tersebut memiliki beberapa persamaan dan perbedaan, yang selanjutnya diuraikan di bawah ini.

Javascript

HTML dirancang untuk menampilkan informasi melalui dokumen pada halaman web. Halaman-halaman yang dihasilkan adalah statis dan terkesan membosankan sehingga ada kebutuhan untuk halaman lebih hidup untuk menjaga kepentingan pengguna. JavaScript diciptakan oleh Netscape Navigator 2.x (akhir 1995) untuk memungkinkan halaman web menjadi lebih interaktif. Meskipun namanya mirip dengan Java akan tetapi hal ini tidak ada kaitannya dengan Microsystems Sun Java. Sun dan Netscape pikir itu akan membantu upaya pemasaran mereka untuk menggunakan nama yang mirip karena Java sangat populer pada saat itu. Bahasa ini awalnya ditujukan untuk scripting sisi server, tetapi versi selanjutnya telah memberikan kemampuan pengembang untuk scripting pada sisi klien yang lebih canggih.

JavaScript bersifat sederhana, ringan, *interpreted,* cross-platform bahasa pemrograman dengan kemampuan berorientasi objek yang dapat ditulis langsung ke dokumen HTML. Dengan memasukkan program JavaScript ke halaman Web, penulis dapat mencapai efek multimedia sederhana seperti *rollover* gambar atau menambah alat yang relatif canggih seperti kalkulator, kalkulator hipotek atau

penanganan kejadian pada *form*. Efek tersebut dapat berjalan di browser Web seperti Netscape Communicator dan Microsoft Internet Explorer. Bahasa JavaScript sendiri distandarisasi oleh Standard ECMA-262 *ECMAScript: A general purpose, cross-platform programming language.*

Versi JavaScript awal adalah versi 1.0. Versi berikutnya yang dirilis selama bertahun-tahun dan termasuk 1.1, 1.2, 1.3 dan 1,5. Versi 1.4 dilaksanakan hanya dalam produk server Netscape.

Inti dari JavaScript

Inti JavaScript berisi kumpulan inti dari obyek, seperti Array, Tanggal(Date) dan Matematika(Math) dan satu set inti dari unsur-unsur bahasa seperti operator, struktur kontrol dan pernyataan. Inti JavaScript diperluas untuk mencakup sisi client dan sisi server versi bahasa dengan melengkapi dengan objek-objek tambahan.

Client-side JavaScript. Inti JavaScript diperpanjang dengan menambahkan objek untuk mengendalikan browser dan Document Object Model nya (DOM). Misalnya, sisi klien ekstensi memungkinkan aplikasi untuk menempatkan elemen pada formulir HTML dan menanggapi peristiwa pengguna seperti klik mouse, input form dan navigasi halaman.

Server-side JavaScript. Inti bahasa diperpanjang dengan menambahkan objek-objek yang relevan untuk menjalankan JavaScript pada sisi server. Misalnya, server-side ekstensi memungkinkan aplikasi untuk berkomunikasi dengan database relasional, memberikan kontinuitas informasi ke aplikasi lain atau melakukan manipulasi file di server. Oleh karena itu, perintah-perintah server-side dapat terhubung ke database relasional dari vendor yang berbeda, berbagi informasi di seluruh pengguna aplikasi, mengakses sistem file di server atau berkomunikasi dengan aplikasi lain melalui LiveConnect dan Java. Halaman HTML dengan server-side JavaScript juga dapat mencakup client-side JavaScript.

Persyaratan

Bahasa JavaScript mudah dipelajari dan digunakan. Tidak ada persyaratan yang mahal untuk menghasilkan kode (aplikasi). Apa yang diperlukan adalah pengetahuan untuk belajar bahasa JavaScript, editor teks bebas seperti Notepad dan browser. Sebagian besar browser menjadikan JavaScript sebagai bahasa standar yang digunakan dalam dokumen HTML sehingga tidak perlu untuk tambahan plug-in.

Fitur JavaScript

- Memfasilitasi user interface yang lebih interaktif
- Memberikan kontrol lebih besar atas pengguna browser
- Membaca dan menulis sisi klien dengan cookie
- Dapat digunakan untuk melakukan perhitungan seperti dalam konverter mata uang, kalkulator hipotek dan lain-lain pada sisi client
- Menghasilkan halaman HTML *on-the-fly* tanpa mengakses web server
- Mendeteksi browser, system operasi, ukuran layar, dan lain-lain
- Memvalidasi nilai yang diinput oleh pengguna pada form
- Dapat digunakan untuk manajemen tanggal dan waktu

Inti JavaScript Dasar

Dalam HTML JavaScript disertakan dalam Tag <script> & </ SCRIPT> atau dapat disimpan sebagai file terpisah dengan ekstensi js..

Pernyataan (Statements)

JavaScript menggunakan pernyataan.

Contoh *Statement*:

Basic seperti pada x = y + z;

Atau Gabungan seperti jika(y == 5) {z += 25}

Sebuah kumpulan pernyataan dikelompokkan bersama-sama untuk dieksekusi yang biasanya disebut sebagai fungsi.

Karakter Set

JavaScript menggunakan set karakter Unicode. Ini berarti bahwa ia menggunakan dua byte untuk mewakili setiap karakter dalam sebuah string. JavaScript adalah case sensitive sehingga programmer harus dapat mengetahui setiap kapitalisasi kode mereka. Secara default, semua perintah JavaScript dalam huruf kecil. Atau, HTML tidak bersifat case sensitive yang bisa menjadi sumber kesalahan untuk halaman HTML. Kata kunci ditulis dgn tanda penghubung dari bahasa lain yang perlu ditangani ditulis dalam C-style: border-right-color seperti dalam CSS, dan bisa ditulis dengan borderRightColor dalam JavaScript.

White Space

JavaScript mengabaikan white space dan jeda baris. Text string di beberapa baris harus digabungkan dengan menggunakan operator (+).

Titik koma

Semua pernyataan JavaScript diakhiri dengan titik koma(;). Jika Anda memiliki satu pernyataan per baris, itu tidak memerlukan titik koma akan tetapi hal ini juga bias sedikit membantu untuk *debugging* dan pemeliharaan. Hal ini juga akan menghilangkan masalah yang mungkin timbul jika pernyataan dijalankan di beberapa baris.

Komentar

Komentar pada JavaScript sama seperti pada C dan C + +.

Gunakan *double slash* (//) untuk komentar satu baris dan / && * * / untuk komentar yang lebih dari satu baris. JavaScript harus berurusan dengan komentar HTML juga.

Meskipun mengakui tanda <! sebagai awal dari sebuah komentar pada HTML, akan tetapi JavaScript tidak melihat tanda(- - >) sebagai tanda penutup. Untuk antisipasi hal tersebut maka dapat digunakan tandan seperti / / &&&& - ->.

Literal

Literal numerik ditulis seperti 125.

String literal harus disertakan dalam tanda kutip yang seimbang misalnya 'JavaScript itu' atau "Ini juga bisa". Tanda kutip ganda dan tunggal dipertukarkan tetapi harus seimbang. Untuk *null* literal tidak memerlukan tanda kutip.

Identifier

Identifier adalah nama yang ditugaskan untuk variabel. Mereka harus diawali dengan huruf, garis bawah atau $. Semua karakter harus karakter ASCII yang valid. Mereka tidak bisa sama sebagai karya JavaScript cadangan.

Masalah Keamanan

Beberapa teknik telah diambil untuk membangun beberapa fitur keamanan JavaScript. Misalnya client-side JavaScript tidak dapat mengakses sumber daya jaringan atau menulis ke *filesystem*. Model Keamanan JavaScript didasarkan pada Java. Secara teori, script download dijalankan secara default dalam lingkungan terbatas 'sandbox' yang mengisolasi mereka dari sisa dari sistem operasi. Naskah diijinkan akses hanya untuk data dalam dokumen saat ini atau dokumen memiliki keterkaitan yang erat (umumnya mereka dari situs yang sama sebagai dokumen saat ini). Tidak ada akses diberikan kepada sistem file lokal, ruang memori dari program yang sedang berjalan lainnya, atau lapisan jaringan sistem operasi. Penahanan semacam ini dirancang untuk mencegah script rusak atau berbahaya (malicious scripts) yang dapat mendatangkan malapetaka di lingkungan

pengguna. Meskipun demikian, beberapa script berbahaya keluar dari sandbox dan melanjutkan untuk menginstal spyware dan trojan. Cara paling aman adalah dengan menonaktifkan scripting, akan tetapi anda juga harus bersedia untuk menerima kenyataan kalau susah menghadirkan situs dinamis pada web browser. Masalah serupa juga bias terjadi pada VBScript.

VB-Script

VBScript (*Visual Basic Scripting*) adalah bahasa scripting yang dirancang untuk memperkenalkan beberapa tingkat interaktivitas di halaman-halaman web. VBScript adalah solusi Microsoft untuk masalah halaman statis. Visual Basic Scripting Edition Microsoft, merupakan versi skala bawah dari Visual Basic untuk digunakan dengan halaman web dan aplikasi lain yang menggunakan Microsoft *Active X* kontrol. Penggunaan sintaks VBScript sangat mirip dengan JavaScript, akan tetapi pada VBScript didasarkan pada subset dari bahasa Visual Basic meskipun tidak menawarkan fungsionalitas dari Visual Basic. Selama scripting diaktifkan pada browser, VBscript dapat ditambahkan ke dokumen HTML. Untuk mendefenisikan VBScript sebagai bahasa scripting maka dapat digunakan tag <script>

<script type="text/vbscript">

Kemiripan antara JavaScript dan VBScript

1. Kedua bahasa yang mudah dipelajari dan tidak memerlukan alat pengembangan yang mahal
2. Keduanya dapat digunakan untuk meningkatkan halaman web
3. JavaScript dan VBScript, berjalan di mesin klien dan dapat menggantikan program CGI untuk mengurangi beban server
4. Keduanya dapat menyalahgunakan dan menjalankan script berbahaya (malicious) pada mesin klien

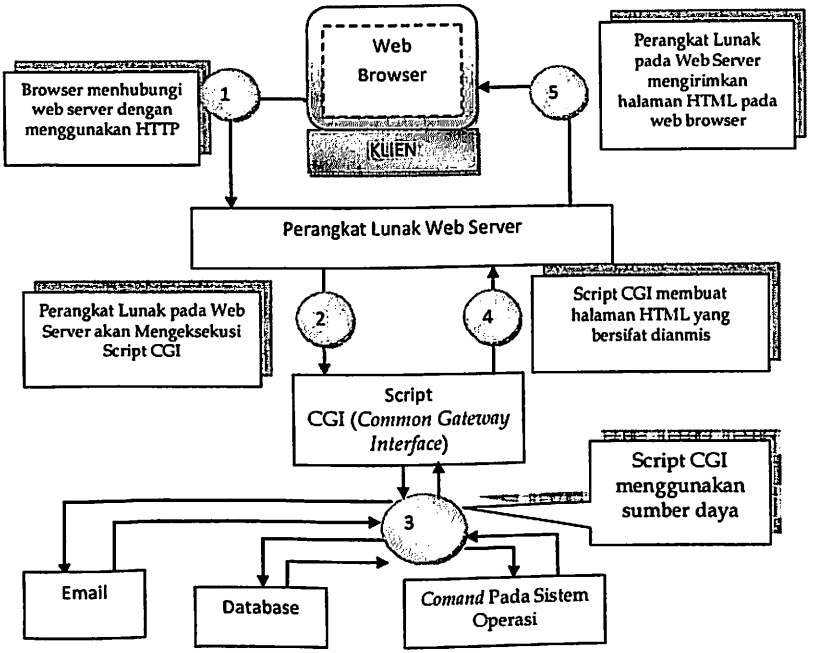
Perbedaan JavaScript dan VBScript

1. JavaScript adalah default bahasa scripting pada beberapa jenis browser, berbeda jenis VBScript yang harus ditetapkan sebagai bahasa scripting.
2. JavaScript memiliki cross-platform dukungan dari semua browser populer sementara VBScript didukung MS IE saja, dengan demikian akan kehilangan pen yang cukup besar.
3. Salah satu masalah yang paling signifikan dengan JavaScript adalah bahwa ada rilis yang berbeda dari bahasa sejak awal(versi 1.0). Demikian, versi yang berbeda dari browser yang ada pada mesin pengguna. Oleh karena itu, kode yang ditulis untuk satu versi belum tentu bekerja pada versi yang lain. Dengan demikian diperlukan pengujian pada tahap pengembangan.
4. JavaScript menggunakan karakter yang sama untuk penggabungan seperti halnya untuk penambahan (karakter +) sedangkan karakter '&' digunakan pada VBScript.

4.4.2 Server Side Scripting

Server-side scripting biasanya digunakan untuk menyediakan sebuah antarmuka dan untuk membatasi akses klien ke database proprietary atau sumber data lainnya. Script ini dapat merakit karakteristik klien untuk digunakan dalam menyesuaikan respon berdasarkan karakteristik, kebutuhan pengguna, hak akses, dan lain-lain. Server-side scripting juga memungkinkan pemilik website untuk mengurangi akses pengguna ke kode sumber. Server-side script dapat ditulis dalam bahasa pemrograman seperti Perl, PHP, ASP.NET, Java dan server-side VBScript yang dieksekusi oleh web server ketika adanya permintaan layanan melalui browser. Server-side script menghasilkan output dalam format yang dapat dimengerti oleh web browser (biasanya HTML), yang kemudian dikirim ke komputer pengguna. Pengguna tidak dapat melihat kode sumber script ini (kecuali penulis

menerbitkan kode terpisah), dan bahkan mungkin tidak menyadari bahwa script dieksekusi. Gambar dibawah ini, menjelaskan proses yang terjadi pada pemrograman berbasis server side scripting.



Gambar 5.2
Transaksi Klien dan Server pada server side scripting

Teknik pada Server-side Scripting untuk WCAG 2.0

Dalam buku ini diangkat beberapa teknik scripting untuk WCAG 2.0 (*Web Content Accessibility Guidelines 2.0*). Penggunaan teknik khusus ini tidak menggantikan teknik umum dimana pengembangan sebuah konten harus mempertimbangkan antara kedua teknik ini, untuk mendapatkan kesesuain dari konten yang dihasilkan. Beberapa teknik tersebut adalah:

- Menerapkan pengalihan otomatis pada sisi server bukan pada sisi client
- Menggunakan htaccess untuk memastikan bahwa satu-satunya cara untuk mengakses konten non-conforming adalah dari sesuai konten.
- Menggunakan HTTP referer untuk memastikan bahwa satu-satunya cara untuk mengakses konten non-conforming adalah dari sesuai konten
- Memungkinkan pengguna untuk memberikan preferensi untuk tampilan sesuai versi alternatif
- Menentukan bahasa default di header HTTP

1. Menerapkan Pengalihan Otomatis Pada Sisi Server Bukan Pada Sisi Client

Teknik ini berkaitan dengan:

- Bagaimana membuat perubahan atas permintaan
- Memahami perubahan atas permintaan

Tujuan dari teknik ini adalah untuk menghindari kebingungan yang mungkin disebabkan ketika dua halaman baru dimuat dalam suksesi cepat karena satu halaman (yang diminta oleh pengguna) diarahkan ke yang lain. Beberapa agen pengguna(*browser*) mendukung penggunaan elemen META pada HTML untuk mengarahkan pengguna ke halaman lain pada detik-detik tertentu. Hal ini membuat halaman tidak dapat diakses oleh beberapa pengguna, terutama pengguna dengan pembaca layar. Teknologi server-side menyediakan metode untuk menerapkan pengalihan dengan cara yang tidak membingungkan pengguna. Sebuah script pada sisi server atau file konfigurasi dapat menyebabkan server untuk mengirim respon HTTP yang sesuai dengan kode status dalam kisaran 3xx dan *header* lokasi dengan URL yang lain. Ketika browser menerima respon ini, perubahan lokasi dan bar browser membuat permintaan dengan URL baru.

Contoh Penggunaan pada beberapa bahasa pemrograman.

✓ **Pada JSP/Servlets**

Pada Java Servlets atau JavaServer Pages (JSP), pengembang dapat menggunakan HttpServletResponse.sendRedirect(String url).

Contoh Kode:

```
...
public void doGet(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException{
...
response.sendRedirect("/newUserLogin.do");
}
```

Kode di atas akan mengirimkan respon dengan kode status 302 ("Ditemukan") dan header lokasi dengan URL baru ke agen pengguna. Hal ini juga memungkinkan untuk mengatur kode status lain dengan response.sendError(int code, String message) dengan salah satu dari konstanta yang didefinisikan pada interface *javax.servlet.http.HttpServletResponse* sebagai kode status.

Contoh Kode:

```
...
public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
...
response.sendError(response.SC_MOVED_PERMANENTLY,
"/newUserLogin.do"); }
```

Aplikasi menggunakan *HttpServletResponse.encodeURL(String url)* untuk menulis ulang URL karena aplikasi tergantung pada sesi, method nya adalah *HttpServletResponse.encodeRedirectURL(String url)* harus digunakan sebagai pengganti, *HttpServletResponse.sendRedirect(String*

url). Hal ini juga memungkinkan untuk menulis ulang URL dengan *HttpServletResponse.encodeURL(String url)* dan kemudian melewatkan URL dengan *HttpServletResponse.sendRedirec(String url)* .

✓ **Pada ASP**

Pada ASP (*Active Server Page*) dengan VBScript, pengembangan dapat dilakukan dengan menggunakan Response. Redirect

Contoh Kode:

Response.Redirect "newUserLogin.asp"

atau

Response.Redirect("newUserLogin.asp")

Kode di bawah ini adalah contoh yang lebih lengkap dengan kode status HTTP tertentu.

Contoh Kode:

Response.Clear

Response.Status = 301

Response.AddHeader "Location", "newUserLogin.asp"

Response.Flush Response.End

✓ Pada PHP

Dalam PHP, pengembang dapat mengirim header HTTP baku dengan menggunakan metode *header method.* Kode di bawah mengirimkan kode status 301 dan lokasi baru. Jika status tidak secara eksplisit diatur, respon redirect mengirimkan kode status HTTP 302.

Contoh Kode:

<?php

header("HTTP/1.1 301 Moved Permanently);

header("Location: http://www.example.com/newUserLogin.php");

?>

2. Menggunakan htaccess

Tujuan dari teknik ini adalah untuk memastikan bahwa pengguna selalu dapat mengakses accessible version dari konten ketika versi non-conforming tidak tersedia. Setiap kali konten disediakan dalam format yang tidak sesuai dengan WCAG, secara keseluruhan situs masih bisa menyesuaikan dengan versi alternatif dari penyediaan konten yang bersifat *inaccessible*. Teknik ini menjelaskan bagaimana penulis dapat menggunakan Modul Apache "mod_access" untuk memastikan bahwa konten non-conforming hanya dapat diakses dari URI .

Contoh 1

sebuah berkas. Htaccess *menggunakan Apache modul* mod_redirect *untuk mengarahkan permintaan "inaccessible.html" menjadi "accessible.html" dengan pengecualian bahwa permintaan sebelumnya bukan berasal dari "accessible.html".*

Contoh Kode:

```
# Jika permintaan untuk konten inaccessible dari sebuah file
# Maka panggil accessible.html, lalu set variabel environment
# Lalu munculkan versi inaccessible
SetEnvIf Referer .*(accessible.html)$ let_me_in
        <FilesMatch ^(inaccessible.html)$>
        Order Deny,Allow
        Deny from all
        Allow from env=let_me_in
</FilesMatch>
# Jika permintaan berasal dari accessible.html
# Maka kirimkan tanggapan dengan kondisi error
# Muncul kode error 403 /nama_direktory/accessible.html
```

Contoh 2

Contoh ini mengasumsikan struktur direktori di mana dokumen yang tersedia dalam berbagai format. Salah satu format tidak memenuhi WCAG dan menggunakan ekstensi file JNA (*Just Not Accessible*). Semua file-file ini disimpan dalam sebuah folder bernama JNA dengan file htaccess.

Contoh Kode:

```
# Jika permintaan untuk konten inaccessible bersumber
  dari file yang berada pada
# http://contoh.com/dokumen/index.html,
# Maka set lingkungan variable untuk menyetujui
  dimunculkannya versi inaccessible.
  SetEnvIf Referer ^http://example.com/documents/
  index.html$ let_me_in
        <FilesMatch ^(.*\.jna)$>
              Order Deny,Allow
              Deny from all Allow from env=let_me_in
        </FilesMatch>
# Jika permintaan berasal dari sumber selain http://
  contoh.com/dokumen/index.html
# Maka kirim response sebagai kondisi error dimana
  lokasi link yang diakses
# ErrorDocument 403 http://contoh.com/dokumen/
  index.html
```

3. Menggunakan HTTP referer

Konten yang dibuat menggunakan *server-side scripting* menyediakan versi yang sesuai(*conforming*) dengan konten sebagai alternatif untuk versi *non-conforming* berdasarkan *HTTP Referer*. Teknik ini berkaitan dengan tingkat kesesuaian dengan persyaratan yang ada. Hal ini disebabkan karena

beberapa agen pengguna tidak mendukung *header referer HTTP*. Tujuan dari teknik ini adalah untuk memastikan bahwa pengguna dapat memperoleh versi *conforming* dari konten. meskipun terdapat versi *conforming* dan *non-conforming*.

Teknik ini menjelaskan cara menggunakan informasi yang diberikan oleh *HTTP referer* untuk memastikan bahwa konten *non-conforming* hanya dapat dicapai dari halaman *conforming*. *Header HTTP referer* diatur oleh agen pengguna dan berisi URI dari halaman (kalau ada) yang disebutkan pada agen pengguna ke halaman *non-conforming*. Untuk menerapkan teknik ini, penulis mengidentifikasi URI untuk versi sesuai konten(*conforming*), pada setiap halaman yang tidak sesuai(*non- conforming*). Ketika permintaan untuk halaman yang memiliki versi *non-conforming* diterima, selanjutnya server membandingkan nilai dari versi *Header HTTP referer* terhadap URI dan melakukan penyesuaian untuk menentukan apakah link ke *versi non-conforming* berasal dari versi *conforming*. Versi *non-conforming* akan dilayani jika *HTTP referer* cocok dengan URI dari versi *non-conforming*.

Contoh Menggunakan Http referer di PHP

Contoh berikut menggambarkan bagaimana teknik ini dapat digunakan dalam PHP. Ini mencakup dua file, conforming.php dan non-conforming.php yang bekerja sama untuk memastikan bahwa satu-satunya cara untuk mengakses konten non-conforming adalah dari konten conforming.

Contoh Kode untuk conforming.php-:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
```

```
<title>Konten Bersifat conforming </title>
</head>
<body>
<h1>Ini adalah halaman yang dapat diakses</h1>
<p>Anda dapat melihat halaman yang non-conforming
<a href="non-conforming.php">di sini</a>. </p>
</body>
</html>
```

Contoh Kode untuk non-conforming.php:

```
<?php
// Jika permintaan(request) yang berasal dari file yang
memuat string
// maka berikan halaman "conforming.php"
if(stristr($_SERVER['HTTP_REFERER'],
"conforming.php"))
{
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>Konten yang bersifat Non-Conforming </title>
</head>
<body>
<h1>Ini adalah sebuah Halaman  non-conforming </h1>
<p>Karena berasal dari <?php echo $_SERVER
['HTTP_REFERER']; ?>, Anda dapat melihat isi
konten pada halaman ini </p>
```

```
</body>
</html>
<?php
}
// Jika gagal menampilkan isi konten pada conforming.php
//maka kembalikan pada pengguna versi conformingnya
else
{
header("Location: conforming.php");
}
?>
```

4. Memungkinkan pengguna untuk memberikan preferensi untuk tampilan sesuai versi alternatif

Memberikan preferensi untuk memungkinkan pengguna melihat sesuai versi alternatif dapat dicapai dengan beberapa cara. Salah satu metode yang umum adalah menyediakan link yang memicu proses server-side yang menetapkan sesi atau cookie dengan menggunakan halaman yang termodifikasi atau mengarahkan pengguna ke versi alternatif. Metode lain termasuk menyediakan pilihan tertentu bagi pengguna yang disimpan sebagai bagian dari informasi login pengguna untuk sistem di mana pengguna masuk untuk mengakses halaman Web atau layanan tertentu.

Pengguna yang membutuhkan versi alternatif akan membutuhkan mekanisme yang disediakan pada halaman *non-conforming* untuk dapat mengakse dan menggunakannya. Mekanisme itu sendiri harus sesuai dengan tingkat aksesibilitas.

Contoh

Contoh 1: Mengatur sesi atau persistent cookie untuk menyimpan preferensi pengguna

Sebuah situs web menawarkan link ke halaman "preferensi" pada halaman dalam situs. Pada halaman ini, ada pilihan untuk melihat suatu versi alternatif dari situs. Preferensi mungkin untuk menampilkan versi situs di mana video yang disertakan di situs menampilkan captioning, atau dapat ditawarkan karena situs utama mengandung masalah kesesuaian aksesibilitas yang ditangani hanya melalui alternatif.

Seorang programmer web dapat memilih untuk menangani preferensi melalui cookie, yang dapat ditangani melalui bahasa scripting server-side seperti PHP.

Halaman preferensi dapat ditawarkan sebagai berikut:

Contoh Kode:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Preferensi Situs</title>
</head>
<body>
<h1>Situs Yang direferensikan</h1>
<form id="form1" name="site_prefs" method="post" action="pref.php">
    <fieldset>
        <legend>Versi Mana yang yang Anda inginkan dari situs ini?</legend>
        <input type="radio" name="site_pref" id="site_pref_reg" value="reg" />
        <label for="site_pref_reg">Versi Utama Situs</label>
```

```
<input type="radio" name="site_pref" id="site_pref_axx"
value="axx" />
<label for="site_pref_axx">Versi Accessibility-conforming</
label>
</fieldset>
</form>
</body>
</html>
```

Form di atas akan dikirimkan ke file ***pref.php*** untuk diproses. Cookie diatur dan dalam contoh sederhana browser pengguna diarahkan ke halaman depan situs.

Contoh Kode pref.php:

```
<?php
if(isset($site_pref))
{
setcookie("site_pref",$site_pref, time() + (86400 * 30));
//set selama 30 hari
header("location: http://www.contoh.com");
//kembalikan ke halaman awal
}
?>
```

Halaman awal pada website (index.php) akan di arahkan untuk mengimplementasikan preferensi pengguna.

Contoh Kode index.php:

```
<?
if(isset($site_pref))
{
    if($site_pref="axx")
    {
```

```
    header("location: ./accessible/index.plip");
    }
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head> ...
```

Untuk web yang menggunakan sistem login, preferensi disimpan dalam database pengguna dan dipanggil oleh script server-side untuk menghasilkan halaman yang dapat dilihat oleh pengguna.

5. Menentukan bahasa default pada header HTTP

Tujuan dari teknik ini adalah untuk memberikan informasi tentang bahasa utama atau bahasa dalam halaman Web, untuk mengidentifikasi para pengunjung konten. Header *Content-Language* dapat berisi daftar satu atau lebih kode bahasa, yang dapat digunakan untuk negosiasi bahasa antara agen pengguna dan server. Jika preferensi bahasa dalam agen pengguna diatur dengan benar, negosiasi bahasa dapat membantu pengguna untuk menemukan versi bahasa konten yang sesuai. Header *Content-Language* tidak berfungsi untuk mengidentifikasi bahasa yang digunakan dalam memproses konten. Bahasa pengolahan konten dapat diidentifikasi dengan menggunakan teknik lain, seperti lang atribut dan xml: lang dalam bahasa markup.

Teknik ini memastikan bahwa bahasa utama dari dokumen, sebagaimana ditentukan misalnya di lang atau xml: lang atribut, tercantum dalam header HTTP *Content-Language*.

Contoh 1: Mengatur bahasa konten pada Java Servlet dan JSP

Pada JSP penulisan script dapat menggunakan *response. setHeader ("Content-Language", lang)* di mana "lang" singkatan tag bahasa (misalnya, "en" untuk bahasa Inggris)

Contoh Kode:

```
...
public void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException
{
...
response.setHeader("Content-Language", "en");
PrintWriter out = response.getWriter();
...
}
```

Contoh 2: Pengaturan konten bahasa PHP

Dalam PHP, programmer dapat mengirim header HTTP yang baku dengan metode header. Contoh berikut menetapkan bahasa ke Bahasa Prancis:

Contoh Kode:

```
<?php
header('Content-language: fr');
...
?>
```

Perbedaan Pada BahasaPemrograman yangberbasis Server Side Scripting

✓ **PHP**

PHP menggunakan campuran interpretasi dan kompilasi untuk memberikan campuran terbaik dari kinerja dan fleksibilitas untuk programmer.

Di belakang layar, PHP mengkompilasi script ke serangkaian instruksi (disebut opcodes) setiap kali diakses. Instruksi ini kemudian dieksekusi satu per satu sampai script berakhir. Hal ini berbeda dari bahasa yang dikompilasi secara konvensional seperti C + + di mana kode dikompilasi ke kode executable asli dan eksekusi yang dijalankan sejak saat itu. Sebaliknya, PHP melakuan re-kompilasi (kompilasi ulang) script setiap kali ada permintaan.

Proses kompilasi ulang mungkin akan terlihat membuang-buang waktu bagi prosesor, akan tetapi sebenarnya tidak semua yang berulang itu kurang bagus karena programmer tidak perlu khawatir lagi ketika membuat perubahan pada skrip. Di sisi lain, banyak script memakan waktu lebih lama untuk mengkompilasi daripada waktu untuk mengeksekusi.

PHP menyediakan umpan balik sangat cepat selama pengembangan. Jika terdapat kesalahan di suatu tempat dalam file, PHP akan menolak untuk mengkompilasi halaman sampai script tersebut benar-benar bersih dari *bug* dan ini memberikan kemudahan untuk menemukan baris yang memuat bug tersebut.

Kecepatan hit kompilasi rutin dibatalkan seluruhnya dengan menggunakan akselerator PHP. Salah satu keunggulan utama menggunakan kode yang interpreted adalah bahwa semua memori yang digunakan oleh script dikelola oleh PHP, dan bahasa secara otomatis membersihkannya setelah setiap naskah selesai dieksekusi. Ini berarti bahwa Anda tidak perlu khawatir tentang menutup link database, membersihkan memori, dan sebagainya, karena PHP sendiri yang akan melakukannya.

✓ **Perl**

Perl sebagai bahasa yang "tertua", banyak dibutuhkanoleh proyek *open source* untuk diinstal demi mendapatkan

akselerasi yang baik dari aplikasi yang mereka kembangkan. Ini mengindikasikan bahwa Perl memiliki keuntungan dari segi fleksibel dan juga memiliki koleksi besar modul yang sudah ditulis untuk itu.

Untuk penulisan script yang bebas dari *bug* pada Perl akan sering terlihat cukup mirip pada PHP. Penyebab utama untuk penampilan berantakan Perl adalah bahwa programmer Perl banyak bergantung pada "one-liners", dimana kemasan sejumlah besar fungsi menjadi hanya satu baris kode. Perl pernah digambarkan sangat akurat oleh penciptanya, Larry Wall, ketika ia berpendapat bahwa sampul depan buku O'Reilly nya pada *Perl should be a camel,* mengatakan bahwa *Perl was ugly but serviceable, and able to go long distances without much nourishment.*

Perl sering merupakan pilihan yang lebih baik ketika ingin mengambil keuntungan dari beberapa library sebelum penulisan script. CPAN, repositori perpustakaan Perl, sangat besar, dan ada sejumlah besar kode untuk yang dapat diambil, diadakan penyesuaian dan digunakan kembali.

✓ **ASP**

Active Server Pages (ASP) dan ASP.NET adalah upaya Microsoft untuk berhasil di pasar pengembangan web dan hadir sebagai standar dengan web server mereka, IIS.

Pengguna dapat menulis add-on untuk ASP menggunakan Visual Basic dan COM akan embuat seluruh solusi yang sangat menarik, akan tetapi faktanya bahwa ASP hanya benar-benar bekerja dengan baik pada IIS.

ASP/ASP.NET umumnya disukai ketika tumpukan semua-Microsoft adalah di tempat. Ketika digunakan pada Windows, sangat mudah untuk menyebarkan. Kode NET ke halaman ASP.NET atau bahkan menulis halaman ASP menggunakan C #.

✓ **JSP**

Java Servlet Pages menggunakan Java, bahasa yang sudah memiliki seperangkat kemampuan yang besar dimana sejumlah besar fungsi yang tersedia. Java juga kondusif untuk skalabilitas karena dapat didistribusikan pada beberapa komputer dengan baik.

Sun, serta anggota masyarakat lainnya, telah bekerja keras untuk mempromosikan bahasa dan alat-alat yang mendukungnya, yang berarti bahwa JSP memiliki banyak dukungan di dalam perusahaan yang lebih besar. Hal ini sangat mendorong terciptanya template halaman untuk kode maksimum digunakan kembali.

JSP adalah pilihan populer ketika ada back-end dari logika bisnis yang ditulis pada java, karena hal ini membuat tim pengembangan bekerja pada bahasa yang homogen.

4.5 Rancangan Tampilan

Keberhasilan atau kegagalan dari suatu situs web tidak sepenuhnya bergantung pada desain visual, karena pengunjung halaman adalah satu-satunya orang yang mengklik mouse dan menentukan segalanya. Desain yang sesuai dengan kebutuhan pengguna utama telah menjadi pendekatan standar untuk desain web yang sukses dan berorientasi pada keuntungan. Dalam buku ini tidak bahas rincian pelaksanaan desain melainkan berfokus pada prinsip-prinsip utama dan pendekatan untuk desain web yang efektif. Pendekatan yang digunakan dengan benar dapat mengarah pada keputusan desain yang lebih canggih dan menyederhanakan proses penyerapan informasi yang disajikan.

Dalam menggunakan prinsip dengan baik dan benar, pertama-tama yang perlu untuk dipahami adalah bagaimana pengguna berinteraksi dengan situs web, bagaimana mereka berpikir dan apa pola dasar perilaku pengguna.

Bagaimana pengguna berpikir?

Pada dasarnya, kebiasaan pengguna di web tidak jauh berbeda dari kebiasaan pelanggan di toko. Pengunjung melirik setiap halaman baru, memindai beberapa teks, dan klik pada link pertama yang menarik minat mereka. Sebagian besar pengguna mencari sesuatu yang menarik atau berguna dan segera diklik akan tetapi jika halaman baru tidak memenuhi harapan pengguna maka kemungkinan besar tombol Back diklik dan proses pencarian dilanjutkan.

Bagaimana Pola Dasar perilaku pengguna?

Pola dasar perilaku pengguna saat berhadapan dengan sebuah website, antara lain:

- **Pengguna menghargai kualitas dan kredibilitas.** Jika halaman website menyediakan konten berkualitas tinggi untuk pengguna, mereka pasti bersedia untuk berkompromi dengan iklan dan desain situs. Ini adalah alasan mengapa tampilan web tidak dirancang dengan baik, situs web dengan konten berkualitas tinggi mendapatkan banyak lalu lintas selama bertahun-tahun karena konten adalah lebih penting daripada desain yang mendukungnya.
- **Pengguna tidak membaca melainkan memindai halaman.** Pengguna lebih suka menganalisis halaman web, mencari beberapa poin tetap atau *anchor* yang akan mengarahkan mereka mendapatkan isi halaman yang sesuai.
- **Pengguna web tidak sabar dan bersikeras pada kepuasan instan.** prinsip Sangat **sederhana.** Jika sebuah situs web tidak dapat memenuhi harapan pengguna, maka desainer gagal mendapatkan minat dari pengguna dan besar kemungkinan perusahaan yang menggunakan jasanya akan merugi. karena bisa dipastikan kalau pengguna akan meninggalkan situs web dan mencari alternatif lain.

- **Pengguna tidak membuat pilihan yang optimal.** Kepuasaan pengguna adalah ketika mereka menemukan pilihan yang masuk akal pada website. Segera setelah mereka menemukan link yang sesuai dengan tujuan, maka besar kemungkinan bahwa link tersebut akan diklik. Jadi, bisa dikatakan bahwa mengoptimalkan web itu sedikit lebih sulit dibandingkan dengan membuat kepuasan pada pengunjung.
- **Pengguna mengikuti intuisi mereka.** Dalam kebanyakan kasus pengguna bukannya membaca informasi yang telah disediakan oleh desainer. Menurut *Steve Krug*, alasan dasar untuk itu adalah bahwa pengguna tidak peduli. *"If we find something that works, we stick to it. It doesn't matter to us if we understand how things work, as long as we can use them. If your audience is going to act like you're designing billboard, then design great billboards."* Bukan suatu masalah yang besar selama tampilan bisa difahami dan bisa digunakan oleh pengguna.
- **Pengguna ingin memiliki kontrol.** Pengguna ingin dapat mengontrol browser mereka untuk penyajian data yang konsisten dari seluruh situs. Misalnya mereka tidak ingin jendela baru muncul tiba-tiba dan mereka ingin bisa mendapatkan kembali dengan tombol "Back" ke situs mereka sebelumnya.

Prinsip utama desain web yang efektif:

1. **Jangan membuat pengguna berpikir**

Menurut hukum pertama Krug kegunaan dari halaman harus web **jelas (*obvious and self-explanatory*)**. Ketika membuat sebuah situs, tugas desainer adalah untuk menyingkirkan *tanda tanya* mengenai pro, kontra dan alternatif.

Jika navigasi dan arsitektur situs yang tidak intuitif, jumlah tanda tanya tumbuh dan membuat lebih sulit bagi pengguna untuk memahami bagaimana sistem bekerja dan

bagaimana untuk mendapatkan informasi dari satu titik A ke titik B. Sebuah struktur yang jelas, petunjuk visual moderat dan mudah dikenali dan link dapat membantu pengguna untuk menemukan apa yang mereka inginkan. Dengan mengurangi beban kognitif, maka desainer membuat pengunjung untuk lebih mudah dan memahami ide di balik sistem. Setelah desainer mencapai hal ini, maka dia dapat mengkomunikasikan mengapa sistem ini berguna dan bagaimana pengguna bisa mendapatkan keuntungan dari itu.

2. Jangan sia-siakan kesabaran pengguna

Dalam setiap proyek ketika Anda akan menawarkan pengunjung Anda beberapa layanan atau alat, cobalah untuk menjaga kebutuhan pengguna. Jangan paksakan pengguna menguji layanan karena pengunjung lebih cenderung acak untuk benar-benar mencobanya. Pertama kali pengunjung bersedia untuk **bermain dengan layanan,** tidak mengisi formulir web yang lama untuk account mereka karena mungkin akun tersebut tidak pernah digunakan di masa depan. Biarkan pengguna menjelajahi situs dan menemukan layanan Anda tanpa memaksa mereka ke berbagi data pribadi. Idealnya **menghapus semua hambatan,** tidak memerlukan langganan atau registrasi terlebih dahulu. Proses pendaftaran awal saja sudah cukup bagi pengguna untuk akses ke website.

3. Mengelola untuk memusatkan perhatian pengguna

Sebagai situs web menyediakan konten baik statis dan dinamis, beberapa aspek dari antarmuka dapat menarik perhatian pengguna. Misalnya penggunaan gambar yang lebih *eye-catching* dari teks, seperti kalimat ditandai dengan huruf tebal tentu lebih menarik daripada teks biasa.

Mata manusia adalah sangat non-linear sehingga **langsung dapat mengenali tepi, pola dan gerakan.** Inilah sebabnya mengapa iklan berbasis video sangat menjeng-

kelkan dan mengganggu, tetapi dari perspektif pemasaran mereka melakukan pekerjaan dengan sempurna untuk menangkap perhatian pengguna.

4. Upayakan Pemaparan fitur website

Desain web modern biasanya dikritik karena pendekatan mereka membimbing pengguna dengan visual menarik1-2-3, adanya langkah demi langkah, tombol besar dengan efek visual dan lain-lain. Tapi dari perspektif desain elemen-elemen sebenarnya bukan hal yang buruk itru dapat mengarahkan pengunjung melalui konten situs dengan cara yang sangat sederhana dan *user-friendly*.

Membiarkan pengguna melihat dengan jelas apa fungsi yang tersedia adalah prinsip fundamental dari desain user interface yang sukses. Yang penting adalah bahwa konten dapat dipahami dengan baik dan pengunjung merasa nyaman dengan cara mereka berinteraksi dengan sistem.

5. Memanfaatkan menulis efektif

Media website berbeda dari media cetak, sehingga perlu penyesuaian gaya penulisan yang sesuai dengan preferensi dan kebiasaan browsing dari pengguna. Menulis promosi tidak akan dibaca. Blok teks panjang tanpa gambar dan kata kunci ditandai dengan huruf **tebal** atau huruf *miring* akan dilewati. Bahasa berlebihan akan diabaikan.

Sebuah solusi optimal untuk menulis efektif dalam website adalah:

- menggunakan frase singkat dan ringkas (datang ke titik secepat mungkin),
- menggunakan tata letak *scannable* (mengkategorikan konten, menggunakan elemen visual dan daftar bullet yang membuat adanya keragaman dalam penulisan),
- menggunakan bahasa sederhana dan obyektif (promosi tidak perlu terdengar seperti iklan akan tetapi website

yang dibangun mampu memberikan alasan yang masuk akal dan objektif sehingga pengunjung merasa harus menggunakan layanan Anda atau tetap pada situs web Anda)

6. **Kesederhanaan**

Prinsip "keep it simple" (KIS) harus menjadi tujuan utama dari desain situs, upayakan untuk kesederhanaan bukan kompleksitas.

7. **Jangan takut dengan white space**

Pentingnya white space tidak hanya membantu **mengurangi beban kognitif** bagi pengunjung, namun memungkinkan untuk melihat informasi yang disajikan di layar. Ketika pengunjung baru melihat desain tata letak pada website, maka hal pertama yang mereka lakukan adalah untuk memindai halaman dan membagi area konten menjadi potongan-potongan informasi yang dapat mereka cerna. Jika Anda memiliki pilihan antara memisahkan dua segmen desain dengan garis terlihat oleh beberapa spasi, itu biasanya lebih baik untuk menggunakan solusi spasi .

8. **Berkomunikasi secara efektif dengan "visible language"**

Pada penulisan Aaron Marcus mengenai komunikasi visual yang efektif dia menyatakan tiga prinsip dasar yang terlibat dalam penggunaan "visible language" *yaitu:*

- **Organize:** menyediakan pengguna dengan struktur konseptual yang jelas dan konsisten. Aturan tersebut hendaknya diterapkan untuk semua elemen.
- **Economize:** Empat poin utama yang harus dipertimbangkan: Kesederhanaan, kejelasan, kekhasan, dan penekanan. *Kesederhanaan* : hanya mencakup elemen-elemen yang paling penting untuk komunikasi, *Kejelasan* : Semua komponen harus dirancang sehingga maknanya

tidak ambigu, *kekhususan* : properti yang paling penting harus bisa dibedakan, *Penekanan* : elemen yang paling penting harus mudah dibedakan dan dirasakan oleh pengunjung.

- **Communicate :** sesuai antara presentasi halaman dengan kemampuan dari pengguna. User interface harus tetap menjaga keseimbangan, mudah dibaca, tipografi, simbolisme dan warna atau tekstur untuk berkomunikasi dengan sukses. Gunakan **maks. 3 tipografi dalam maksimum 3 ukuran titik** - maksimal 18 kata atau 50-80 karakter per baris teks.

9. Konvensi

Desain konvensional elemen situs tidak menghasilkan situs web membosankan. Bahkan, **konvensi ini sangat berguna** karena mereka mengurangi kurva belajar, kebutuhan untuk mengetahui bagaimana sesuatu bekerja. Misalnya, penggunaan layanan RSS-feed, itu akan menjadi sesuatu hal yang aneh semua situs web memiliki presentasi visual yang berbeda. Hal ini sebenarnya tidak jauh berbeda dari kehidupan manusia dimana kita cenderung terbiasa dengan prinsip-prinsip dasar misalnya bagaimana kita mengatur data (folder) atau melakukan belanja (penempatan produk). Dengan konvensi Anda bisa mendapatkan kepercayaan, kehandalan dan membuktikan kredibilitas Anda. **Ikuti harapan pengguna dengan** memahami apa yang mereka harapkan dari navigasi, struktur teks, penempatan fungsi pencarian dan lain-lain. (untuk informasi lebih lanjut, lihat Nielsen Alertbox Usability pada http://www.useit.com/alertbox/).

10. Pengujian

Proses pengujian harus diterapkan pada setiap proyek desain web karena pengujian akan memberikan **pengetahuan penting** mengenai masalah yang signifikan dan isu-isu yang berkaitan dengan tata letak tertentu. Prinsip dasar uji (*testing*) aplikasi:

- **Pengujian dengan menggunakan satu pengunjung adalah jauh lebih baik dari pada pengujian** tanpa adanya user, pengujian awal dalam proyek ini adalah lebih baik daripada pengujian 50 mendekati akhir.
- Pengujian adalah **proses berulang-ulang.** Itu berarti bahwa desain yang dibuat harus diuji, diperbaiki dan kemudian diuji kembali. Mungkin ada masalah yang belum ditemukan selama putaran pertama sebagai pengguna praktis diblokir oleh masalah lain.
- Uji kegunaan **selalu menghasilkan hasil yang bermanfaat.** Entah akan menunjuk ke masalah yang sudah ada atau akan menunjuk kepada adanya cacat desain utama.
- **Pengembang dan desainer tidak cocok untuk menguji. Ada banyak alasan yang masuk akal kenapa pengembang dan desainer tidak dituntut untuk melakukan pengujian pada website yang dibangun hal ini dapat disebabkan karena** mereka sudah tahu bagaimana website itu dibangun, bagaimana cara kerjanya serta adanya faktor keletihan setelah mengembangkan situs.

Catatan:
Jika Anda ingin situs yang besar, Anda harus mengujinya.

4.6 Gaya Lebar Bahasa

4.6.1 Cascading Style Sheets (CSS)

CSS adalah bahasa style sheet yang digunakan untuk menggambarkan tampilan dan format dari dokumen yang ditulis dalam bahasa markup. Penerapannya paling umum adalah penambahan gaya pada halaman web yang ditulis dalam HTML dan XHTML. Tetapi CSS juga dapat diterapkan untuk setiap jenis dokumen XML, SVG dan XUL. CSS

dirancang untuk memungkinkan adanya pemisahan antara konten dokumen dengan gaya tampilan dokumen, termasuk unsur-unsur seperti tata letak, warna dan huruf (font). Pemisahan ini dapat meningkatkan aksesibilitas konten, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi karakteristik presentasi, memungkinkan beberapa halaman untuk berbagi format dan mengurangi kompleksitas dan pengulangan dalam konten structural. CSS juga dapat memungkinkan halaman markup yang sama disajikan dalam gaya yang berbeda untuk metode rendering yang berbeda, hal ini juga bergantung pada ukuran layar atau perangkat yang sedang dilihat. CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna *mouseover*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya.

Versi CSS

Pada saat penulisan buku ini terdapat beberapa versi CSS yaitu CSS1, CSS2, CSS2.1, CSS3 dan CSS4. CSS1 dikembangkan berpusat pada pemformatan dokumen HTML, CSS2 dikembangkan untuk memenuhi kebutuhan terhadap format dokumen agar bisa ditampilkan di printer, sedangkan CSS3 adalah versi terbaru dari CSS yang mampu melakukan banyak hal dalam desain website. CSS2 mendukung penentuan posisi konten, downloadable, huruf font, tampilan pada tabel/table layout dan media tipe untuk printer. Kehadiran versi CSS yang kedua diharapkan lebih baik dari versi pertama. CSS3 juga dapat melakukan animasi pada halaman website, diantaranya animasi warna hingga animasi 3D. Dengan CSS3 desainer lebih dimudahkan dalam hal kompatibilitas websitenya pada smartphone dengan dukungan fitur baru yakni media query. Selain itu, banyak fitur baru pada CSS3 seperti: multiple background, border-

radius, drop-shadow, border-image, CSS Math, dan CSS Object Model. CSS memiliki berbagai tingkat dan profil. Setiap tingkat dibangun berdasarkan CSS yang terakhir, biasanya menambahkan fitur baru dan biasanya dinotasikan sebagai CSS 1, CSS 2, CSS 3, dan CSS 4. Profil biasanya subset dari satu atau lebih tingkat dari CSS, saat ini ada profil untuk perangkat mobile, printer, dan televisi. Profil tidak boleh berbeda dengan berbagai jenis media, yang ditambahkan dalam CSS 2.

1. CSS 1

Spesifikasi CSS pertama yang menjadi rekomendasi W3C resmi CSS tingkat 1, diterbitkan pada bulan Desember 1996. Di antara kemampuan adalah dukungan untuk:

- Properti Font seperti jenis huruf dan penekanan
- Warna teks, latar belakang, dan elemen lainnya
- Teks atribut seperti spasi antara kata, huruf dan baris teks
- Keselarasan dari teks, gambar, tabel dan elemen lainnya
- Margin, border, padding dan positioning untuk tiap elemen
- Memberikan identifikasi dan klasifikasi yang unik pada kelompok atribut

---

Catatan:
W3C tidak lagi mempertahankan CSS 1 Rekomendasi.

---

2. CSS 2

Spesifikasi untuk CSS level 2 dikembangkan oleh W3C dan diterbitkan sebagai rekomendasi pada Mei 1998. Sebuah superset dari CSS 1, CSS 2 mencakup sejumlah kemampuan baru seperti posisi absolut, relatif dan tetap elemen dan z-index, konsep jenis media, dukungan untuk style sheet au-

ral dan teks dua arah dan properti font baru seperti bayangan.

Catatan:

W3C tidak lagi mempertahankan CSS 2 rekomendasi.

---

3. CSS 2.1

CSS level 2 revisi 1 sering disebut sebagai "CSS 2.1", merupakan perbaikan kesalahan dalam CSS 2, menghilangkan fitur yang kurang didukung atau tidak sepenuhnya oleh spesifikasi browser. Setelah dikaji oleh Komite Penasehat W3C, akhirnya CSS 2.1 diterbitkan sebagai Rekomendasi W3C pada tanggal 7 Juni 2011.

Catatan:

Hanya Internet Explorer 8 mendukung penuh CSS 2.1

---

4. CSS 3

Berbeda dengan CSS 2, dimana spesifikasi tunggal mendefinisikan berbagai fitur, maka spesifikasi dari CSS 3 dibagi menjadi beberapa dokumen terpisah yang disebut "modul". Setiap modul menambahkan kemampuan baru atau memperluas fitur yang didefinisikan pada CSS 2, karena bersifat modularisasi maka modul yang berbeda memiliki stabilitas dan status yang berbeda. Beberapa modul (termasuk Backgrounds and Borders dan Tata letakc multikolom) memiliki status *Calon Rekomendasi (CR)* dan dianggap cukup stabil. CSS 3 juga mendukung untuk penambahan tepi bulat untuk elemen melalui properti border-radius.

Catatan:

Pada saat penulisan buku ini, banyak programmer menggunakan style ini pada website yang mereka bangun untuk tujuan estetika.

---

5. CSS 4

W3C mulai menyusun CSS 4 pada 29 September 2009. Namun, saat ini tidak didukung oleh sebagian besar web browser.

Sintaks CSS

CSS memiliki sintaks sederhana dan menggunakan sejumlah kata kunci dalam bahasa Inggris untuk menentukan nama properti. Sebuah style sheet terdiri dari daftar *aturan*. Setiap aturan terdiri dari satu atau lebih *penyeleksi* dan satu *blok deklarasi*. Setiap deklarasi itu sendiri terdiri dari *properti*, tanda titik dua (:) dan *nilai*. Jika ada multiple deklarasi di blok sebuah semi-colon (;) harus dimasukkan untuk memisahkan deklarasi masing-masing. Dalam CSS, *penyeleksi* digunakan untuk menyatakan bagian mana dari style yang berlaku.

Contoh peulisan(kode) CSS sebagai berikut:

```
p {
   color: #006600;
}
```

Pada kode di atas **p** adalah selector, isi yang berada diantara tanda { } adalah *declaration* sementara **color** adalah *property* dan **#006600** adalah *value* dari property tersebut.

Metode Penulisan CSS

Ada beberapa teknik dalam penulisan CSS, yaitu:

Inline Style Sheet

Style CSS didefinisikan langsung pada elemen HTML yang bersangkutan. Cara penulisannya cukup dengan menambahkan atribut **style="..."** dalam elemen HTML tersebut. Style hanya akan berlaku pada elemen yang bersangkutan, dan tidak akan memengaruhi elemen HTML yang lain. Contoh penulisan CSS dengan metode inline style sheet:

```
<html>
<head>
<title>Contoh Bentuk Inline </title>
</head>
<body bgcolor="#FFFFFF">
<p id="p1" style="font-size:20pt">
Elemen P ini diformat dengan besar font 20 point </p>
<p id="p2" style="font-size:14pt; color:red">
Elemen P ini diformat dengan besar font 14 point dan
menggunakan warna merah </p>
</body>
</html>
```

Embedded Style Sheet

CSS didefinisikan terlebih dahulu dalam tag **<style> ... </style>** di atas elemen **<body>**. Pada pendefinisian ini disebutkan atribut-atribut CSS yang akan digunakan untuk elemen-elemen HTML, yang selanjutnya dapat digunakan gaya(style) elemen yang bersangkutan.

Contoh penggunaan CSS dengan metode embedded style sheet:

```
<html>
<head>
<title>Contoh Bentuk Embedded</title>
</head>
<style>
body {background:#0000FF; color:#FFFF00; margin-
left:0.5in}
h1 {font-size:18pt; color:#FF0000}
p {font-size:12pt; font-family:arial; text-indent:0.5in}
```

```
</style>
<body>
<h1>Judul ini berukuran 18 dengan warna merah!</h1>
<p>Elemen p ini di format dengan besar font 12 point
dengan tipe font Arial dan mempunyai identasi 0.5 inch
</p>
</body>
</html>
```

Style ditulis pada Halaman Terpisah (Eksternal Style)

Penulisan style CSS didefinisikan pada halaman terpisah dengan menyimpan file css tersebut dengan ekstensi css. Style tersebut selanjutnya dipanggil pada halaman HTML dengan menggunakan fungsi link dan ditempatkan dalam elemen head.

Contoh penulisan untuk pemanggilan file css:

```
<link rel = "stylesheet" href = "http://contoh.com/style.css"
type = "text / css" />
```

Contoh Penggunaan style CSS tersebut adalah:

a. Buat file baru dan simpan dengan nama style.css

```
body {
    background:#0000FF;
    color:#FFFF00;
    margin-left:0.5in
    }
h1 {
    font-size:18pt;
    color:#FF0000
    }
p{ font-size:12pt; font-family:arial; text-indent:0.5in }
```

b. Buat file HTML dengan memanggil file yhang di buat pada poin a.

```
<html>
<head>
<title>Contoh Penggunaan Style CSS untuk eksternal style</title>
<link rel = "stylesheet" href = "http://contoh.com/style.css" type = "text / css" />
</head>
<body>
<h1>Judul ini berukuran 18 dengan warna merah!</h1>
<p>Elemen p ini di format dengan besar font 12 point dengan tipe font Arial dan mempunyai identasi 0.5 inch </p>
</body>
</html>
```

Prioritas CSS

Berdasarkan metode penulisan style CSS di atas, maka dalam sebuah halaman HTML memungkinkan untuk menerapkan metode penulisan lebih dari satu. Dari penggunaan metode penulisan yang beragam inilah, maka aturan (prioritas) style ditentukan.

Prioritas skema CSS (dari tertinggi ke prioritas terendah) adalah:

- Inline style
- Embedded Style Sheet
- Style sheet eksternal file CSS terpisah dirujuk dari dokumen lain

Style sheet dengan prioritas tertinggi mengontrol tampilan konten. Penentuan style untuk sebuah halaman diatur dari prioritas tertinggi diteruskan ke sumber prioritas terendah proses inilah yang disebut *cascading*.

4.6.2 XLS

XSL adalah Stylesheet yang khusus dikembangkan sebagai komplemen XML, untuk merubah informasi pada XML ke dalam bentuk lain agar bisa ditampilkan di layar, dicetak pada kertas ataupun di dengar oleh manusia. Pada dasarnya proses ini di bagi menjadi dua bagian proses yakni Proses Transformasi Struktural yang meliputi pengumpulan, pengelompokan dan pengurutan data maupun penyusunan ulang, penambahan dan penghapusan elemen dan atribut, dan yang kedua adalah Proses Merubah Format menjadi pixel pada layar, nohtah tinta di kertas atau nada di speaker. Proses yang pertama itulah yang kemudian disebut XSLT(*Extensible Stylesheet Language Transformations*), sedangkan yang kedua biasa disebut XSLFO (*eXtensible Stylesheet Language:Formatting Object*). XSLT digunakan untuk mengkonversi data XML ke HTML atau XHTML dokumen untuk ditampilkan sebagai halaman web . Transformasi dapat terjadi secara dinamis baik pada klien maupun pada sisi server. Hal ini juga digunakan untuk membuat output untuk mencetak atau menampilkan video langsung, biasanya dengan mengubah XML asli ke *XSL Formatting Objects* untuk membuat output diformat yang kemudian dapat dikonversi ke berbagai format, termasuk PDF, PostScript dan PNG. XSLT juga dapat menerjemahkan pesan XML atau membuat perubahan pada dokumen dalam lingkup skema tunggal, misalnya dengan menghapus bagian-bagian dari pesan yang tidak diperlukan.

Sebenarnya untuk menampilkan dokumen XML agar lebih menarik dilihat di browser bisa dilakukan oleh Cascade StyleSheet. CSS yang sering digunakan untuk memformat HTML bisa juga dipakai untuk XML. Akan tetapi CSS tidak mampu melakukan tugas tugas yang rumit seperti memformat angka desimal, menjumlah, menghitung rata-rata, menampilkan gambar, dan lain-lain. Dan untuk melakukan tugas-tugas itulah diperlukan XSLT.

Model pengolahan pada XSLT melibatkan:

- satu atau lebih dokumen XML
- satu atau lebih modul XSLT *stylesheet*
- template XSLT *prosesor* dan
- satu atau lebih dokumen *hasil* pengolahan.

XSLT Prosesor biasanya mengambil dua dokumen masukan yaitu dokumen XML dan modul XSLT stylesheet. Stylesheet XSLT berisi kumpulan instruksi aturan template yang akan digunakan oleh XSLT prosesor untuk menghasilkan dokumen output. XSLT Processor atau yang biasa disebut Parser adalah software bantu yang tugasnya menerapkan perintah-perintah dalam XSLT pada dokumen sumber XML, dan menghasilkan dokumen keluaran baik berupa HTML, Text file ataupun XML. Bila browser yang digunakan oleh pengguna adalah Internet Explorer versi 5.5 ke bawah, secara default menggunakan MSXML atau MSXML2 sebagai Processor yang dibuat oleh Microsoft.

Catatan:

Implementasi beberapa browser seperti Google Chrome tidak mendukung XSLT bila diterapkan ke file lokal hal ini untuk pertimbangan keamanan, terutama dikaitkan dengan kerentanan di Gmail.

4.7 Teknologi Multimedia

Flash memiliki keuntungan karena menjadi lebih terkenal dari Microsoft Silverlight akan tetapi hal ini juga menjadi keuntungan untuk Microsoft Silverlight, karena banyak pengguna Flash ingin melihat perbedaan dari kedua teknologi tersebut seperti optimasi mesin pencari, animasi, pengolahan suara, kompatibilitas platform dan lain-lain. Adapun fitur-fitur yang tidak didukung oleh flash tersebut banyak

dikembangkan oleh Silverlight. Para desain web tentu memiliki pilihan untuk membuat keputusan mengenai teknologi apa yang sesuai dengan kebutuhan mereka. Beberapa alasan yang patut dipertimbangkan ketika ingin menggunakan flash atau silverlight dalam desain dan implementasi sistem berbasis web, adalah sebagai berikut:

- **Animasi**

Flash dan Silverlight mampu menciptakan animasi yang berkualitas. Jadi mana yang lebih baik untuk digunakan? Silverlight menggunakan model animasi WPF(*Windows Presentation Foundation),* yang berbeda dengan frame pada animasi berbasis Flash. Di sisi lain flash menggunakan model animasi berbasis frame. Sebuah objek diciptakan untuk setiap frame dari animasi, sehingga efek animasi menggunakan frame demi frame. Waktu dapat menjadi masalah, kecuali track audio tertanam kosong digunakan untuk menjaga frame rate yang konstan.

- **Ukuran file**

XAML(**Extensible Application Markup Language) pada** Silverlight adalah non terkompresi, sehingga deskripsi bahasa ini cenderung lebih besar dari format file Flash yang terkompresi.

Flash menggunakan format file yang terkompresi, sehingga menghasilkan file yang relatif kecil, dengan gambar dan teks yang tertanam dalam film.

- **Scripting**

Scripting untuk Flash dan Silverlight, sangat berbeda.

Silverlight dapat ditulis dalam beberapa bahasa pemrograman termasuk Visualbasic.net, dan Visual C #. Net. Javascript, C # dan VB.Net dapat digunakan untuk *client side scripting* yang berjalan pada kerangka .NET.

ActionScript pada Flash adalah bahasa pemrograman berorientasi objek yang dapat menggunakan back-end dari teknologi lain yang menggunakan bahasa pemrograman dan *framework* seperti Ruby on rails, ASP dan PHP. Flash memilki *class library* untuk mengembangkan aplikasi berbasis desktop ataupun aplikasi browser secara online. Sehingga memungkinkan untuk memiliki berbagai macam control dalam merancang antarmuka pengguna.

- **Video dan Audio**

Keduanya dapat memanfaatkan fitur ini, akan tetapi pada buku ini akan membahas bagaimana mereka melakukannya dan format apa yang digunakan.

Pada Silverlight, format yang digunakan adalah format standar pada windows seperti WMA dan WMV dengan *VC-1 Codec* untuk video. Silverlight mendukung *SDK(Software Development Kit) encoder* yang gratisan, jika seseorang tidak memiliki movie maker, dalam kasus seseorang tidak memiliki pembuat film. *SDK encoder* memberikan kesempatan bagi pengembang untuk mengkodekan secara manual.

Flash menggunakan beberapa format video, untuk mendukung teknologi ini flash mengembangkan *codec* terbaru dengan mengoptimalkan penggunaan bandwidth serta kualitas video yang sangat tinggi.

- **Pemrosesan Suara**

Silverlight memiliki dukungan dari .Net untuk pemutaran dan hasil dari proses tersebut ditampilkan pada browser dan API tidak mendukung untuk audio tingkat rendah. Oleh karena itu, file dengan ekstensi WAV tidak dapat dimainkan pada silverlight.

Actionscript pada Flash, memungkinkan suara ditambahkan ke film yang sedang diputar, sehingga suara untuk seluruh film dapat dikontrol.

- **Aksesibilitas**

Silverlight 3 dapat memberikan akses untuk semua sistem warna, yang berarti kontrol sistem operasi dapat diakses sebagian besar orang sehingga membuat aplikasi lebih mudah dibaca.

Flash kaya akan aksesibilitas fitur serta memberi kemudahan kepada para penyandang cacat. Misalnya keterangan yang terdapat pada video sangat berguna bagi orang-orang yang memilki masalah dengan pendengaran. Pengembangan aplikasi berbasis flash, kompatibel dengan layar dan perangkat lain sejenisnya yang membuatnya berfungsi penuh dengan *shortcut* pada keyboard. Hal ini memungkinkan kontrol pemutaran standar seperti play, maju, mundur dan lain-lain yang dapat dimanfaatkan sepenuhnya pada keyboard. Flash sedikit lebih maju dengan memungkinkan fungsi-fungsi ini menjadi tab, sehingga membuat banyak orang mudah menggunakannya.

- **Kompatibilitas.**

Sistem operasi apa yang mendukung kedua teknologi tersebut?

Silverlight dapat berjalan pada sistem operasi Windows Vista/XP/2000, Windows Server 2003/2008, Windows Mobile 6, Mac OS 10.1/10.5 (PowerPC) dan Mac OS 10.1/10.5 (Intel) dan tidak mendukung Linux dan Solaris.

Flash mendukung Windows Vista/XP/2000, Windows Server 2003/2008, Mac OS 10.1/10.5 (PowerPC), Mac OS 10.1/10.5 (Intel), Linux 5, openSUSE 11, Ubuntu 7.10 atau yang lebih baru dan Solaris 10.

- **Teks dan SEO**

Silverlight memperlakukan teks sebagai entitas yang terpisah pada web server dan kemudian aplikasi sepenuhnya dicari dan diindeks oleh mesin pencari. Antarmuka pengguna

yang dinyatakan dalam XAML dan diprogram dalam kerangka. NET. Annimation dan grafis vektor dapat ditandai dengan menggunakan XAML.

Meskipun flash yang tradisional belum bersifat SEO friendly akan tetapi Adobe bekerja sama dengan Google dan Yahoo telah memiliki cara untuk membuat aplikasi berbasis flash dapat diindex dan ditemukan pada *search engine*. Saat ini Google telah membuat kemajuan dalam mengindeks aplikasi Flash.

- **Format gambar.**

File apa yang didukung oleh kedua teknologi tersebut? Silverlight mendukung utamanya JPEG dan format file PNG. Format lain yang didukung tetapi umumnya dengan keterbatasan tertentu.

Flash mendukung hampir semua format file.

- **Pemrograman Socket.**

Silverlight mendukung komunikasi socket domain silang antara aplikasi Silverlight dan server apapun. Silverlight mendukung pemrograman soket melalui namespace *System.Net.Sockets*, dan mendukung pengiriman data bolak-balik melintasi soket pada port, mulai port 4502-4534.

Flash menggunakan metode yang berbeda. Objek XML Socket mengimplementasikan soket klien yang memungkinkan komputer yang menjalankan Flash untuk berkomunikasi dengan komputer server diidentifikasi oleh alamat IP atau nama domain.

- **Dukungan Pada Webcam**

Apakah Silverlight dan Flash mendukung webcam? Silverlight, sayangnya tidak mendukung webcam atau mikrofon.

Flash mendukung webcam dan mikrofon untuk video langsung dengan menggunakan perintah tertentu.

- **Penyebaran Aplikasi**

Menyebarkan Silverlight merupakan urusan yang cukup kompleks, dimana masing-masing komponen membutuhkan penyebaran secara terpisah. Setiap permintaan web dari Silverlight membutuhkan komponen-komponen berikut yaitu:

- File XML
- File DLL (jika perlu)
- File Silverlight.js
- File JavaScript lain,
- Sumber Daya (gambar, audio, video).

Flash memiliki semua gambar, animasi dan teks, dikompresi ke dalam satu file *shock wave file* (SWF). Ini terlihat sedikit sederhana dibandingkan dengan Silverlight. Akan tetapi karena sifatnya yang terkompresi inilah yang membuat file flash sulit untuk diindeks dan ditemukan oleh *search engine*. Satu-satunya downside, adalah karena sifatnya dikompresi, itu jauh lebih sulit untuk mencari dan indeks mesin pencari.

- **Aplikasi**

Bisakah memainkan film sebagai aplikasi?

Silverlight, tidak mendukung pemutaran film sebagai aplikasi.

Flash dapat memutar film pada desktop serta film di flash dapat dikompilasi sebagai file eksekusi dan bermain di aplikasi yang berdiri sendiri.

- **Media Streaming**

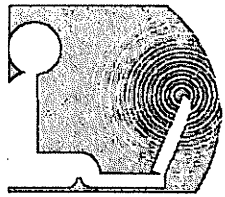
Silverlight, memiliki layanan tersendiri untuk streaming media dalam bentuk *Microsoft Silverlight Streaming* oleh *Windows Live*. Hal ini membuat lebih mudah bagi pengembang untuk memberikan media yang kaya untuk aplikasi mereka. Hal ini juga memungkinkan mereka untuk streaming atau

host antar browser, interaktif dan media konten untuk windows atau mac. Perangkat pada pihak ketiga lainnya seperti *Microsoft Expression Studio,* juga dapat digunakan bersama dengan layanan ini untuk membuat dan mengembangkan konten yang interaktif.

Flash, sayangnya tidak menyediakan layanan untuk host.{"}

Catatan:

Flash, jelas pilihan yang lebih baik jika Anda perlu aplikasi untuk berjalan pada sistem operasi seperti Linux atau Solaris, tetapi jika Anda membutuhkan aplikasi yang akan diindeks oleh search engine, maka mungkin Silverlight adalah pilihan yang lebih baik. Akan tetapi pilihan teknologi yang akan digunakan dalam desain dan implementasi website akan kembali bergantung pada kebutuhan dan kemampuan sumber daya yang dimiliki.



Daftar Pustaka

Andrei Dragoi, Octavian. *The Conceptual Architecture of the Apache Web Server,* University of Waterloo, Department of Computer Science, 1999.

Cecchet, Emmanuel. *Building PetaByte Warehouses with Unmodified PostgreSQL,* 2011.

Copper, I. dkk. *Web Caching & Replication (RFC 3040),* University of Maryland, Baltimore County, 2008.

D. Davison, Brian.*Predicting Web Actions from HTML Content,* Computer Science & Packard Lab-United States of America : Engineering Dept. Lehigh University.

Fette, I. dkk. *The Websocket Protocol,* Internet Engineering Task Force(RFC 6455), 2011.

Fielding, R. dkk. *Hypertext Transfer protocol – HTTP/1.1,* Network working group request for Comments : 2616 ,1999.

Gordon B. Davis, *Management Information System: Conceptual Foundation, Structure, and Development,* McGraw-Hill International Book Company, Auckland, 1974.

H. Pence, James. *How to Do Everything with HTML & XHTML,* California : Corel VENTURA Publisher, 2003.

Hadisutopo, Ariesto. *Pemrograman Flash Dengan PHP dan MySQL,* Yogyakarta: Graha Ilmu, 2007.

Hammer-Lahav, E. dkk. *The OAuth 2.0 Protocol draft-ietf-oauth-v2-02,* Internet-Draft, 2010.

Hermawan, Julius. *Analisa dan Desain Pemrograman Berorientasi Objek.* Yogyakarta: Andi Offset, 2004.

Kadir, Abdul. *Penuntun Praktis Belajar SQL*. Yogyakarta : Penerbi

Kay, Michael (2008). *XSLT 2.0 and XPath 2.0 Programmer* Wile, 2008.

Kementrerian Komunikasi dan Informatika Republik *Panduan Keamanan Web Server*, 2011.

Krishnamurthy, B. dkk. *Web Protocols and Practice : HTTP 1.1, Protocols, Caching, and Traffic Measurement*. Addison-W

Marshall, James. *A Practical Guide to Writing Clients and Se*

Monroe, R. dkk. *Architectural Styles, Design Patterns, and O* Software, 1997.

Nixon, Robin. *Learning PHP, MySQL, and JavaScript*, Unit America : O'Reilly Media, Inc., 2009.

Nugroho, Adi. *Konsep Pengembangan Sistem Basis Data*. Penerbit Informatika, 2004.

O'neil, Patrick dkk. *Database Principles, Programming, and I Second Edition*, United States of America San Fran demic press, 2001.

Paoli, Jean dkk . *Extensible Markup Language (XML) 1.0 (Fi* 2008.

Pfaffenberger, Brian dkk. *HTML, XHTML, and CSS Bible,* Indianapolis, Indiana: Wiley Publishing, Inc., 2004

Rob, Peter dkk. *Database system : Design, Implementation, a ment. Eighth Edition*, Thompson Place, Boston Ma United States of America. 2009.

S.Sudarma. *Panduan belajar MySQL Database Server*, Jakarta : me

Schwartz, Baron dkk . *High Performance MySQL*, Unite America : O'Reilly media, 2008.

Stalling, William. *Komunikasi Data dan Komputer edisi 8(Data and computer communications, 8th ed.)*, Jakarta : Salemba Infotek, 2011.

Suarga, Faisal dkk. *Pengantar Teknologi Informasi 1*, Makassar: Alauddin Press, 2006.

Suhendi, Edi. *Kreatif Dengan Adobe Flash Profesional*, Bandung: Yama Wydia, 2009.

Tidwell, Jennifer. *Designing Interfaces: Patterns for Effective Interaction Design*, United States of America : OReilly Media, 2005.

Vaswani, Vikram. *How Do Everything With PHP and MySQL*, United States of America: McGraw-Hill Companies, 2005.

Zaki, Ali dan SmitDev Community. *36 Menit Belajar Komputer PHP dan MySQL*, Jakarta: Elex Media Komputindo, 2008.

Biodata Penulis

Nama lengkap **Ridwan Andi Kambau**, lahir pada tanggal 5 Januari 1974 di kota Makassar. Tamat Sekolah Dasar pada tahun 1986, disekolah SD Pertiwi Gunung Sari Baru. Setelah lulus dari Sekolah Dasar, selanjutnya pada tahun yang sama masuk di SMP Negeri 1 Makassar dan lulus pada tahun 1989. Sekolah Menengah Atas(SMA) ditamatkan pada tahun 1992.

Pada tahun 1999, menyelesaikan program pendidikan sarjana(S1) jurusan/Program Studi Teknik Elektro di Universitas Hasanuddin. Lulus Program Magister(S2), pada tahun 2006 di Universitas Indonesia dengan jurusan teknologi Informasi. Sebagai bentuk pengabdian kepada bangsa dan Negara maka pada tahun 2008, mendaftar dan terangkat sebagai dosen di Universitas Islam Negeri Alauddin Makassar.

Pada tahun 2010 mengeluarkan jurnal Teknosains, yang berjudul Konsep Pengembangan *"Ubiquitous Computing"* Era Baru Teknologi Informasi dan Komunikasi. Pada tahun 2009-2012 terangkat sebagai Kepala Pusat Informasi dan Komputer. Pada tahun 2012 ini pula, mendapatkan penghargaan *"The Most Committed Public University In ICT Implementation"* dari Gamatechno Indonesia.{"}

ISBN 602-237-370-0

ALAUDDIN UNIVERSITY PRESS
082348671117 - au_press@yahoo.com